Multivariate GARCH estimation via a Bregman-proximal trust-region method

Stéphane Chrétien[1] and Juan-Pablo Ortega[2]

Abstract

The estimation of multivariate GARCH time series models is a difficult task mainly due to the significant overparameterization exhibited by the problem and usually referred to as the "curse of dimensionality". For example, in the case of the VEC family, the number of parameters involved in the model grows as a polynomial of order four on the dimensionality of the problem. Moreover, these parameters are subjected to convoluted nonlinear constraints necessary to ensure, for instance, the existence of stationary solutions and the positive semidefinite character of the conditional covariance matrices used in the model design. So far, this problem has been addressed in the literature only in low dimensional cases with strong parsimony constraints (see for instance [ASPL03] for the diagonal three-dimensional VEC handled with ad-hoc techniques). In this paper we propose a general formulation of the estimation problem in any dimension and develop a Bregman-proximal trust-region method for its solution. The Bregman-proximal approach allows us to handle the constraints in a very efficient and natural way by staying in the primal space and the Trust-Region mechanism stabilizes and speeds up the scheme. Preliminary computational experiments are presented and confirm the very good performances of the proposed approach.

Key Words: multivariate GARCH, VEC model, volatility modeling, multivariate financial time series, Bregman divergences, Burg's divergence, LogDet divergence, constrained optimization.

1 Introduction

Autoregressive conditionally heteroscedastic (ARCH) models [Eng82] and their generalized counterparts (GARCH) [Bol86] are standard econometric tools to capture the leptokurticity and the volatility clustering exhibited by financial time series. In the one dimensional situation, a large collection of parametric models that account for various stylized features of financial returns is available. Additionally, adequate model selection and estimation tools have been developed, as well as explicit characterizations of the conditions that ensure stationarity or the existence of higher moments.

One of the advantages of GARCH models that makes them particularly useful is that once they have been calibrated they provide an estimate of the dynamical behavior of volatility which, in principle, is not directly observable. This feature makes desirable the extension of the GARCH prescription to the multivariate case since such a generalization provides a dynamical picture of the correlations between different assets which are of major importance, in the context of financial econometrics, for pricing and hedging purposes, asset allocation, and risk management in general.

[1]Département de Mathématiques de Besançon, , Probability and Statistics Group, Université de Franche-Comté, UFR des Sciences et Techniques. 16, route de Gray. F-25030 Besançon cedex. France. Stephane.Chretien@univ-fcomte.fr

[2]Centre National de la Recherche Scientifique, Département de Mathématiques de Besançon, , Probability and Statistics Group, Université de Franche-Comté, UFR des Sciences et Techniques. 16, route de Gray. F-25030 Besançon cedex. France. Juan-Pablo.Ortega@univ-fcomte.fr

This generalization is nevertheless not free from difficulties. The most general multivariate GARCH models are the VEC prescription proposed by Bollerslev *et al* [BEW88] and the BEKK model by Engle *et al* [EK95]; both families of models present satisfactory properties that match those found in univariate GARCH models, nevertheless their lack of parsimony, even in low dimensions makes them extremely difficult to calibrate; for example, VEC(1,1) models require $n(n+1)(n(n+1)+1)/2$ parameters, where n is the dimensionality of the modeling problem; BEKK(1,1,1) requires $n(5n+1)/2$. Indeed, due to the high number of parameters needed, it is rare to find these models at work beyond two or three dimensions and even then, ad hoc estimation techniques are used and additional limitations are imposed on the model to make it artificially parsimonious; see for example [ASPL03] for an illustration of the estimation of a three dimensional DVEC model (VEC model with diagonal parameter matrices [BEW88]) using constrained non-linear programming. These difficulties have lead to the search for more parsimonious but still functioning models like for example CCC [Bol90], DCC [TT02, Eng02] or GDC [KN98]. On a different vein, a number of different signal separation techniques have been tried out in the financial time series context with the aim of reducing this intrinsically multivariate problem to a collection of univariate ones. For example, principal component analysis is used in the O-GARCH model [Din94, AC97, Ale98, Ale03] and independent component analysis in the ICA-GARCH model [WYL06, GFGPP08]. We advice the reader to check with the excellent reviews [BLR06, ST09] for a comprehensive description of these and other models.

Despite the overparameterization problem we will concentrate in this work on full fledge VEC models. This decision is taken not for the pure sake of generality but because the intrinsic difficulties of this parametric family of models make them an ideal benchmark for testing optimization techniques subjected to potentially complex matrix constraints. Stated differently, it is our belief that, independently from the pertinence of the VEC family in certain modeling situations, any optimization algorithm developed to estimate them will work smoothly when applied to more elementary situations. Hence, the work that we present in this paper is capable of increasing the range of dimensions in which VEC models can be estimated in practice by improving the existing technology in two directions:

- Explicit matrix formulation of the model and of the associated stationarity and positivity constraints: the works in the literature usually proceed by expressing the constraints in terms of the entries of the parameter matrices (see for example [ASPL03] in the DVEC case). A global matrix formulation is necessary in order to obtain a dimension independent encoding of the problem. This task is carried out in Sections 2 and 3

- Use of a Bregman-type proximal optimization algorithm that efficiently handles the constraints in the primal space. More specifically, we will be using **Burg's matrix divergence**; this divergence is presented, for example, in [KSD09a] and it is a particular instance of a **Bregman divergence**. Bregman divergences are of much use in the context of machine learning (see for instance [DT07, KSD09b] and references therein). In our situation we have opted for this technique as it allows for a particularly efficient treatment of positive definiteness constraints, as those in our problem, avoiding the need to solve additional secondary optimization problems that appear, for example, had we used Lagrange duality. It is worth emphasizing that even though the constraints that we handle in the estimation problem admit a simple and explicit conic formulation well adapted to the use of Lagrange multipliers, the associated dual optimization problem is in this case of difficulty comparable to that of the primal so avoiding this extra step is a major advantage. This approach is presented in Sections 4.1 and 4.2. In Section 4.3 we couple the use of Bregman divergences with a refinement of the local penalized model using quadratic BFGS type terms and with a trust-region iteration acceptance rule that greatly stabilizes the primal trajectory and improves the convergence speed of the algorithm. Finally, given the non-linear non-convex nature of estimation via quasi-loglikelihood optimization, the availability of good preliminary estimation techniques is of paramount importance in order to avoid local minima; this point is treated in Section 4.4 where

some of the simpler modeling solutions listed above are used to come up with a starting point to properly initialize the optimization algorithm.

In Section 5 we illustrate the estimation method proposed in Section 4 with various numerical experiments that prove its applicability and support the following statements:

- The trust-region correction speeds up the algorithm and the BFGS modification makes the convergence rate dimensionally independent.
- More importantly, VEC seems to be a performing modeling tool for stock market log-returns when compared with other more parsimonious parametric families, *even in dimensions where the high number of parameters in comparison with the sample size would make us expect a deficient modeling behavior*. Our conjecture is that this better than expected results have to do with the spectral sparsity (in the dimensions we work on we should rather say spectral concentration) of the correlation matrices exhibited by stock market log-returns; this empirically observed feature imposes nonlinear constraints on the model parameters that invalidate the hypotheses necessary to formulate the standard results on the asymptotic normality of the quasi-loglikelihood parameter estimator (see later on expressions (4.1) and (4.2)) and make it more favorable with respect to its use with standard sample sizes. In a forthcoming publication we plan to provide a detailed study of the convergence and complexity properties of the proposed algorithm, together with dimension reduction techniques based on the use of the spectral sparsity that, as we said, is empirically observed in actual financial time series.

Notation and conventions: In order to make the reading of the paper easier, most of the proofs of the stated results have been gathered at the end in the form of appendices. All along the paper, bold symbols like $\mathbf{r}$ denote column vectors, $\mathbf{r}^T$ denotes the transposed vector. Given a filtered probability space $(\Omega, \mathbb{P}, \mathcal{F}, \{\mathcal{F}_t\}_{t\in\mathbb{N}})$ and X, Y two random variables, we will denote by $E_t[X] := E[X|\mathcal{F}_t]$ the conditional expectation, $\mathrm{cov}_t(X,Y) := \mathrm{cov}(X,Y|\mathcal{F}_t) := E_t[XY] - E_t[X]E_t[Y]$ the conditional covariance, and by $\mathrm{var}_t(X) := E_t[X^2] - E_t[X]^2$ the conditional variance. A discrete-time stochastic process $\{X_t\}_{t\in\mathbb{N}}$ is predictable when X_t is $\mathcal{F}_{t-1}$-measurable, for any $t \in \mathbb{N}$.

2 Preliminaries on matrices and matrix operators

Matrices: Let $n, m \in \mathbb{N}$ and denote by $\mathbb{M}_{n,m}$ the space of $n \times m$ matrices. When $n = m$ we will just write $\mathbb{M}_n$ to refer to the space of $n \times n$ square matrices. Unless specified otherwise, all the matrices in this paper will contain purely real entries. The equality $A = (a_{ij})$ denotes the matrix A with components $a_{ij} \in \mathbb{R}$. The symbol $\mathbb{S}_n$ denotes the subspace of $\mathbb{M}_n$ that contains all symmetric matrices

$$\mathbb{S}_n = \{A \in \mathbb{M}_m \mid A^T = A\}$$

and $\mathbb{S}_n^+$ (respectively $\mathbb{S}_n^-$) is the cone in $\mathbb{S}_n$ containing the positive (respectively negative) semidefinite matrices. The symbol $A \succeq 0$ (respectively $A \preceq 0$) means that A is positive (respectively negative) semidefinite.

We will consider $\mathbb{M}_{n,m}$ as an inner product space with the pairing

$$\langle A, B\rangle = \mathrm{trace}(AB^T) \tag{2.1}$$

and denote by $\|A\| = \langle A, A\rangle^{\frac{1}{2}}$ the associated Frobenius norm. Given a linear operator $\mathcal{A} : \mathbb{M}_{n,m} \to \mathbb{M}_{p,q}$ we will denote by $\mathcal{A}^* : \mathbb{M}_{p,q}^* \to \mathbb{M}_{n,m}^*$ its adjoint with respect to the inner product (2.1).

The vec, vech, mat, and math operators and their adjoints: The symbol $\text{vec} : \mathbb{M}_n \to \mathbb{R}^{n^2}$ denotes the operator that stacks all the columns of a matrix into a vector. Let $N = \frac{1}{2}n(n+1)$ and let $\text{vech} : \mathbb{S}_n \to \mathbb{R}^N$ be the operator that stacks only the lower triangular part, including the diagonal, of a symmetric matrix into a vector. The inverse of the vech (respectively vec)operator will be denoted by $\text{math} : \mathbb{R}^N \to \mathbb{S}_n$ (respectively $\text{mat} : \mathbb{R}^{n^2} \to \mathbb{M}_n$).

Given $n \in \mathbb{N}$ and $N = \frac{1}{2}n(n+1)$, let $S = \{(i,j) \in \{1,\dots,n\} \times \{1,\dots,n\} \mid i \geq j\}$ we define $\sigma : S \to \{1,\dots,N\}$ as the map that yields the position of component $(i,j), i \geq j$, of any symmetric matrix in its equivalent vech representation. The symbol $\sigma^{-1} : \{1,\dots,N\} \to S$ will denote its inverse and $\widetilde{\sigma} : \{1,\dots,n\} \times \{1,\dots,n\} \to \{1,\dots,N\}$ the extension of σ defined by:

$$\widetilde{\sigma}(i,j) = \begin{cases} \sigma(i,j) & i \geq j \\ \sigma(j,i) & i < j. \end{cases} \tag{2.2}$$

The proof of the following result is provided in the Appendix.

Proposition 2.1 *Given $n \in \mathbb{N}$ and $N = \frac{1}{2}n(n+1)$, let $A \in \mathbb{S}_n$ and $m \in \mathbb{R}^N$ arbitrary. The following identities hold true:*

(i) $\langle \text{vech}(A), m\rangle = \frac{1}{2}\langle A + \text{diag}(A), \text{math}(m)\rangle$.

(ii) $\langle A, \text{math}(m)\rangle = 2\langle \text{vech}(A - \frac{1}{2}\text{diag}(A)), m\rangle$,

where $\text{diag}(A)$ *denotes the diagonal matrix obtained out of the diagonal entries of A. Let* $\text{vech}^* : \mathbb{R}^N \to \mathbb{S}_n$ *and* $\text{math}^* : \mathbb{S}_n \to \mathbb{R}^N$ *be the adjoint maps of* vech *and* math*, respectively, then:*

$$\text{math}^*(A) = 2\,\text{vech}\left(A - \frac{1}{2}\text{diag}(A)\right), \tag{2.3}$$

$$\text{vech}^*(m) = \frac{1}{2}\left(\text{math}(m) + \text{diag}(\text{math}(m))\right). \tag{2.4}$$

The operator norms of the mappings that we just introduced are given by:

$$\|\text{vech}\|_{op} = 1 \tag{2.5}$$

$$\|\text{math}\|_{op} = \sqrt{2} \tag{2.6}$$

$$\|\text{vech}^*\|_{op} = 1 \tag{2.7}$$

$$\|\text{math}^*\|_{op} = \sqrt{2} \tag{2.8}$$

$$\|\text{diag}\|_{op} = 1 \tag{2.9}$$

Block matrices and the Σ operator: let $n \in \mathbb{N}$ and $B \in \mathbb{M}_{n^2}$. The matrix B can be divided into n^2 blocks $B_{ij} \in \mathbb{M}_n$ and hence its components can be labeled using a blockwise notation by referring to the (k,l) element of the (i,j) block as $(B_{ij})_{kl}$. This notation makes particularly accessible the interpretation of B as the coordinate expression of a linear endomorphism of the tensor product space $\mathbb{R}^n \otimes \mathbb{R}^n$. Indeed if $\{\mathbf{e}_1,\dots,\mathbf{e}_n\}$ is the canonical basis or $\mathbb{R}^n$, we have

$$B(\mathbf{e}_i \otimes \mathbf{e}_k) = \sum_{j,l=1}^{n} (B_{ij})_{kl}(\mathbf{e}_j \otimes \mathbf{e}_l). \tag{2.10}$$

Definition 2.2 *Let $A \in \mathbb{M}_N$ with $N = \frac{1}{2}n(n+1)$. We define $\Sigma(A) \in \mathbb{S}_{n^2}$ blockwise using the expression*

$$\begin{cases} \textit{If } k \geq l \quad (\Sigma(A)_{kl})_{ij} = \begin{cases} \frac{1}{2}A_{\sigma(k,l),\sigma(i,j)}, & \textit{if} \quad i > j \\ A_{\sigma(k,l),\sigma(i,j)}, & \textit{if} \quad i = j \\ \frac{1}{2}A_{\sigma(k,l),\sigma(j,i)}, & \textit{if} \quad i < j \end{cases} \\ \textit{If } k \leq l \quad \Sigma(A)_{kl} = \Sigma(A)_{lk}, \end{cases} \tag{2.11}$$

where σ is the map defined above that yields the position of component $(i,j), i \geq j$, of any symmetric matrix in its equivalent vech representation. By construction $(\Sigma(A)_{kl})_{ij}$ is symmetric with respect to transpositions in the (k,l) and (i,j) indices; this implies that $\Sigma(A)$ is both symmetric and blockwise symmetric. We will refer to any matrix in $\mathbb{S}_{n^2}$ with this property as n**-symmetric** *and will denote the corresponding space by $\mathbb{S}^n_{n^2}$.*

The proofs of the next two results are provided in the Appendix.

Proposition 2.3 *Given $H \in \mathbb{S}_n$ and $A \in \mathbb{M}_N$, with $N = \frac{1}{2}n(n+1)$, the n-symmetric matrix $\Sigma(A) \in \mathbb{S}^n_{n^2}$ that we just defined satisfies:*

$$A \operatorname{vech}(H) = \operatorname{vech}(\Sigma(A) \bullet H), \tag{2.12}$$

where $\Sigma(A) \bullet H \in \mathbb{S}_n$ is the symmetric matrix given by

$$(\Sigma(A) \bullet H)_{kl} = \langle \Sigma(A)_{kl}, H \rangle = \operatorname{trace}(\Sigma(A)_{kl} H).$$

Proposition 2.4 *Let $\Sigma : \mathbb{M}_N \to \mathbb{M}_{n^2}$ be the operator defined in the previous proposition, $N = \frac{1}{2}n(n+1)$. Then, for any $\mathcal{B} \in M_{n^2}$, the corresponding dual map $\Sigma^* : \mathbb{M}_{n^2} \to \mathbb{M}_N$ is given by*

$$\Sigma^*(\mathcal{B}) = 2B - \widetilde{B}, \tag{2.13}$$

where $B, \widetilde{B} \in \mathbb{M}_N$ are the matrices defined by

$$B_{pq} = ((\mathbb{P}^n_{n^2}(\mathcal{B}))_{\sigma^{-1}(p)})_{\sigma^{-1}(q)}, \quad \text{and} \quad \widetilde{B}_{pq} = B_{pq}\delta_{\mathrm{pr}_1(\sigma^{-1}(p)),\mathrm{pr}_2(\sigma^{-1}(p))}.$$

The symbol $\mathbb{P}^n_{n^2}(\mathcal{B})$ denotes the orthogonal projection of $\mathcal{B} \in \mathbb{M}_{n^2}$ onto the space $\mathbb{S}^n_{n^2}$ of n-symmetric matrices that we spell out in Lemma 7.1. As we saw in Proposition 2.3, Σ maps into the space $\mathbb{S}^n_{n^2}$ of symmetric matrices; let $\widetilde{\Sigma} : \mathbb{M}_N \to \mathbb{S}^n_{n^2}$ be the map obtained out of Σ by restriction of its range. The map $\widetilde{\Sigma}$ is a bijection with inverse $\widetilde{\Sigma}^{-1} : \mathbb{S}^n_{n^2} \to \mathbb{M}_N$ given by

$$\left(\widetilde{\Sigma}^{-1}(B)\right)_{p,q} = \left(B_{\sigma^{-1}(p)}\right)_{\sigma^{-1}(q)} \left(2 - \delta_{\mathrm{pr}_1(\sigma^{-1}(q)),\mathrm{pr}_2(\sigma^{-1}(q))}\right). \tag{2.14}$$

3 The VEC-GARCH model

Consider the n-dimensional conditionally heteroscedastic discrete-time process $\{\mathbf{z}_t\}$ determined by the relation

$$\mathbf{z}_t = H_t^{1/2}\boldsymbol{\epsilon_t} \qquad \text{with} \qquad \{\boldsymbol{\epsilon_t}\} \sim \mathrm{IIDN}(\mathbf{0}, \boldsymbol{I}_n).$$

In this expression, $\{H_t\}$ denotes a predictable matrix process, that is for each $t \in \mathbb{N}$, the matrix random variable H_t is $\mathcal{F}_{t-1}$-measurable, and $H_t^{1/2}$ is a square root of H_t, hence it satisfies $H_t^{1/2}(H_t^{1/2})^T = H_t$. In these conditions it is easy to show that the conditional mean $E_t[\mathbf{z}_t] = \mathbf{0}$ and that the conditional covariance matrix process of $\{\mathbf{z}_t\}$ coincides with $\{H_t\}$.

Different prescriptions for the time evolution of the conditional covariance matrix $\{H_t\}$ determine different vector conditional heteroscedastic models. In this paper we will focus on the **VEC-GARCH model** (just VEC in what follows). This model was introduced in [BEW88] as the direct generalization of the univariate GARCH model [Bol86] in the sense that every conditional variance and covariance is a function of all lagged conditional variances and covariances as well as all squares and cross-products of the lagged time series values. More specifically, the VEC(q,p) model is determined by

$$\boldsymbol{h}_t = \mathbf{c} + \sum_{i=1}^{q} A_i \boldsymbol{\eta}_{t-i} + \sum_{i=1}^{p} B_i \boldsymbol{h}_{t-i},$$

where $\boldsymbol{h}_t := \mathrm{vech}(H_t)$, $\boldsymbol{\eta}_t := \mathrm{vech}(\mathbf{z}_t\mathbf{z}^T)$, $\mathbf{c}$ is a N-dimensional vector, with $N := n(n+1)/2$ and $A_i, B_i \in \mathbb{M}_N$.

In the rest of the paper we will restrict to the case $p = q = 1$, that is:

$$\left\{ \begin{array}{rcl} \mathbf{z}_t & = & H_t^{1/2}\boldsymbol{\epsilon_t} \qquad \text{with} \qquad \{\boldsymbol{\epsilon_t}\} \sim \mathrm{IIDN}(\mathbf{0}, \boldsymbol{I}_n), \\ \boldsymbol{h}_t & = & \mathbf{c} + A\boldsymbol{\eta}_{t-1} + B\boldsymbol{h}_{t-1}. \end{array} \right. \tag{3.1}$$

In this case the model needs $N(2N+1) = \frac{1}{2}(n^2+n)(n^2+n+1)$ parameters for a complete specification.

3.1 Positivity and stationarity constraints

The general prescription for the VEC model spelled out in (3.1) does not guarantee that it has stationary solutions. Moreover, as we saw above, the resulting matrices $\{H_t\}_{t\in\mathbb{N}}$ are the conditional covariance matrices of the resulting process and therefore, additional constraints should be imposed on the parameter matrices $\mathbf{c}, A$, and B in order to ensure that they are symmetric and positive semidefinite. Unlike the situation encountered in the one-dimensional case, necessary and sufficient conditions for positivity and stationarity seem very difficult to find and we will content ourselves with sufficient specifications.

Positivity constraints: we will use the sufficient conditions introduced by Gourieroux in [Gou97] that, as we show in the next proposition, can be explicitly formulated using the map Σ introduced in Definition 2.2.

Proposition 3.1 *If the parameter matrices* $\mathbf{c}, A,$ *and* B *in (3.1) are such that* $\mathrm{math}(\mathbf{c}), \Sigma(A),$ *and* $\Sigma(B)$ *are positive semidefinite then so are the resulting conditional covariance matrices* $\{H_t\}_{t\in\mathbb{N}}$*, provided the initial condition* H_0 *is positive semidefinite.*

Second order stationarity constraints: Gourieroux [Gou97] has stated sufficient conditions in terms of the spectral radius of $A + B$ that we will make more restrictive in order to ensure the availability of a formulation in terms of positive semidefiniteness constraints.

Proposition 3.2 *The VEC model specified in (3.1) admits a unique second order stationary solution if all the eigenvalues of* $A + B$ *lie strictly inside the unit circle. This is always the case whenever the top singular eigenvalue* $\sigma_{\max}(A + B)$ *of* $A + B$ *is smaller than one or, equivalently, when the matrix* $\mathbb{I}_N - (A + B)(A + B)^T$ *is positive definite. If any of these conditions is satisfied, the marginal variance of the model is given by*

$$\Gamma(0) = \mathrm{math}(E[\mathbf{h}_t]) = \mathrm{math}((\mathbb{I}_N - A - B)^{-1}\mathbf{c}). \tag{3.2}$$

3.2 The likelihood function, its gradient, and computability constraints

Given a sample $\mathbf{z} = \{\mathbf{z}_1, \ldots, \mathbf{z}_T\}$, the quasi-loglikelihood associated to (3.1) is:

$$\log L(\mathbf{z}; \boldsymbol{\theta}) = -\frac{TN}{2}\log 2\pi - \frac{1}{2}\sum_{t=1}^{T}\log(\det H_t) - \frac{1}{2}\sum_{t=1}^{T}\mathbf{z}_t^T H_t^{-1}\mathbf{z}_t \tag{3.3}$$

where $\boldsymbol{\theta} := (\mathbf{c}, A, B)$. In this expression, the matrices H_t are constructed out of $\boldsymbol{\theta}$ and the sample $\mathbf{z}$ using the second expression in (3.1). This implies that the dependence of $\log L$ on $\boldsymbol{\theta}$ takes place through the matrices H_t. Notice that these matrices are well defined once initial values H_0 and $\mathbf{z}_0$ have been fixed. This initial values are usually taken out of a presample; if this is not available it is customary to take

the mean values associated to the stationary model, namely $\mathbf{z} = \mathbf{0}$ and $H_0 = \mathrm{math}((\mathbb{I}_N - A - B)^{-1}\mathbf{c})$ (see (3.2)). Once the initial conditions have been fixed, it can be shown by induction that

$$\mathbf{h}_t = \left(\sum_{i=0}^{t-1} B^i\right)\mathbf{c} + \sum_{i=0}^{t-1} B^i A \boldsymbol{\eta}_{t-i-1} + B^t \mathbf{h}_0. \tag{3.4}$$

The maximum likelihood estimator $\widehat{\boldsymbol{\theta}}$ of $\boldsymbol{\theta}$ is the value that maximizes (3.3) for a given sample $\mathbf{z}$. The search of that extremal is carried out using an optimization algorithm that we will discuss later on in the paper and that requires the gradient $\nabla_{\boldsymbol{\theta}} \log L(\mathbf{z};\boldsymbol{\theta})$ of $\log L$. In order to compute it we write the total quasi-loglikelihood as a sum of T conditional loglikelihoods

$$l_t(\mathbf{z}_t; A, B, c) = -\frac{N}{2}\log 2\pi - \frac{1}{2}\log(\det H_t) - \frac{1}{2}\mathbf{z}_t^T H_t^{-1}\mathbf{z}_t$$

A lengthy calculation shows that:

$$\nabla_{\mathbf{c}} l_t = \left[(\gamma_t - \Gamma_t)^T \sum_{i=0}^{t-1} B^i\right]^T, \tag{3.5}$$

$$\nabla_A l_t = \left[\sum_{i=0}^{t-1} \boldsymbol{\eta}_{t-i-1}(\gamma_t - \Gamma_t)^T B^i\right]^T, \tag{3.6}$$

$$\nabla_B l_t = \left[\sum_{i=0}^{t-1}\left[\sum_{j=0}^{i-1} B^j(\mathbf{c} + A\boldsymbol{\eta}_{t-i-1})(\gamma_t - \Gamma_t)^T B^{i-j-1} + B^j \mathbf{h}_0 (\gamma_t - \Gamma_t)^T B^{t-j-1}\right]\right]^T, \tag{3.7}$$

where

$$\Gamma_t := \frac{1}{2}\mathrm{math}^*(H_t^{-1}), \quad \gamma_t := \frac{1}{2}\mathrm{math}^*(\Lambda_t), \quad \text{and} \quad \Lambda_t := H_t^{-1}\mathbf{z}_t\mathbf{z}_t^T H_t^{-1}.$$

These formulas for the gradient were obtained by using the explicit expression of the conditional covariance matrices (3.4) in terms of the sample elements and the coefficient matrices. Such a closed form expression is not always available as soon as the model becomes slightly more complicated; for example, if one adds to the model (3.1) a drift term like in [Dua95] for the one dimensional GARCH case, an expression like (3.4) ceases to exist. That is why, in the next proposition, we introduce an alternative iterative method that can be extended to more general models, it is well adapted to its use under the form of a computer code and, more importantly, suggests the introduction of an additional estimation constraint that noticeably shortens the computation time needed for its numerical evaluation.

Proposition 3.3 *Let $\mathbf{z} = \{\mathbf{z}_1, \ldots, \mathbf{z}_T\}$ be a sample, $\boldsymbol{\theta} := (\mathbf{c}, A, B)$, and let $\log L(\mathbf{z};\boldsymbol{\theta})$ be the quasi-loglikelihood introduced in (3.3). Then, for any component θ of the three-tuple $\boldsymbol{\theta}$, we have*

$$\nabla_{\boldsymbol{\theta}} \log L = \sum_{t=1}^{T} \nabla_{\boldsymbol{\theta}} l_t = \sum_{t=1}^{T} T^*_{\boldsymbol{\theta}} H_t \cdot \nabla_{H_t} l_t, \quad \textit{where} \tag{3.8}$$

$$\nabla_{H_t} l_t = -\frac{1}{2}\left[H_t^{-1} - H_t^{-1}\mathbf{z}_t\mathbf{z}_t^T H_t^{-1}\right], \tag{3.9}$$

and the differential operators $T^*_\theta H_t$ are determined by the recursions:

$$T^*_{\mathbf{c}} H_t \cdot \Delta = \mathrm{math}^*(\Delta) + T^*_{\mathbf{c}} H_{t-1} \cdot \mathrm{vech}^*(B^T \mathrm{math}^*(\Delta)), \tag{3.10}$$

$$T^*_A H_t \cdot \Delta = \mathrm{math}^*(\Delta) \cdot \boldsymbol{\eta}_{t-1}^T + T^*_A H_{t-1} \cdot \mathrm{vech}^*(B^T \mathrm{math}^*(\Delta)), \tag{3.11}$$

$$T^*_B H_t \cdot \Delta = \mathrm{math}^*(\Delta) \cdot \mathrm{vech}(H_{t-1})^T + T^*_B H_{t-1} \cdot \mathrm{vech}^*(B^T \mathrm{math}^*(\Delta)), \tag{3.12}$$

with $\boldsymbol{\eta}_t = \mathrm{vech}(\mathbf{z}_t\mathbf{z}^T)$, $\Delta \in \mathbb{S}_n$ *and setting* $T^*_{\mathbf{c}}H_0 = \mathbf{0}$, $T^*_A H_0 = T^*_B H_0 = \mathbf{0}$. *The operators* $T^*_\theta H_t$ *constructed in (3.10)–(3.12) are the adjoints of the partial tangent maps* $T_{\mathbf{c}}H_t : \mathbb{R}^N \to \mathbb{S}_n$, $T_A H_t : M_N \to \mathbb{S}_n$, *and* $T_B H_t : M_N \to \mathbb{S}_n$ *to* $H_t(\mathbf{c}, A, B) := \mathrm{math}(\mathbf{h}_t(\mathbf{c}, A, B))$, *with* $\mathbf{h}_t(\mathbf{c}, A, B)$ *as defined in (3.4).*

Matrix expression of the recursions (3.10)–(3.12): the use of Proposition 3.3 requires translating the operator recursions (3.10)–(3.12) into matrix recursions. In this particular case this can be achieved by writing $\Delta \in \mathbb{S}_n$ as $\Delta = \mathrm{vech}^*(\mathbf{v})$, with $\mathbf{v} = \mathrm{math}^*(\Delta) \in \mathbb{R}^N$. With this change of variables, the expression (3.10) becomes

$$T^*_{\mathbf{c}}H_t \cdot \mathrm{vech}^*(\mathbf{v}) = \mathbf{v} + T^*_{\mathbf{c}}H_{t-1} \cdot \mathrm{vech}^*(B^T\mathbf{v}). \tag{3.13}$$

Let $c_t \in \mathbb{M}_N$ be the matrix associated to the linear operator $T^*_{\mathbf{c}}H_t \circ \mathrm{vech}^* : \mathbb{R}^N \to \mathbb{R}^N$. In view of (3.13), the matrices $\{c_t\}_{t\in\{1,\dots,T\}}$ are determined by the recursions

$$c_t = \mathbb{I}_N + c_{t-1}B^T. \tag{3.14}$$

Once the family $\{c_t\}_{t\in\{1,\dots,T\}}$ has been computed, it can be used in (3.8) by noticing that

$$T^*_{\mathbf{c}}H_t \cdot \Delta = c_t \cdot \mathrm{math}^*(\Delta).$$

Regarding (3.11), let $A_t \in \mathbb{M}_{N^2,\,N}$ be the matrix associated to the linear operator $\mathrm{vec} \circ T^*_A H_t \circ \mathrm{vech}^* : \mathbb{R}^N \to \mathbb{R}^{N^2}$. Given that

$$\mathrm{vec}(\mathbf{v}\boldsymbol{\eta}_{t-1}^T) = \mathrm{vec}(\mathbb{I}_N\mathbf{v}\boldsymbol{\eta}_{t-1}^T) = \left(\boldsymbol{\eta}_{t-1} \otimes \mathbb{I}_N\right)\mathrm{vec}(\mathbf{v}) = \left(\boldsymbol{\eta}_{t-1} \otimes \mathbb{I}_N\right)\mathbf{v},$$

the recursion (3.11) implies that the family $\{A_t\}_{t\in\{1,\dots,T\}}$ is determined by

$$A_t = \left(\boldsymbol{\eta}_{t-1} \otimes \mathbb{I}_N\right) + A_{t-1}B^T \tag{3.15}$$

and hence, once it has been computed, it can be used in (3.8) by noticing that

$$T^*_A H_t \cdot \Delta = \mathrm{mat}\left(A_t \cdot \mathrm{math}^*(\Delta)\right),$$

where we recall that mat denotes the inverse of the vec operator. Finally, let $B_t \in \mathbb{M}_{N^2,\,N}$ be the matrix associated to the linear operator $\mathrm{vec} \circ T^*_B H_t \circ \mathrm{vech}^* : \mathbb{R}^N \to \mathbb{R}^{N^2}$. The family $\{B_t\}_{t\in\{1,\dots,T\}}$ is determined by

$$B_t = (\mathrm{vech}(H_{t-1}) \otimes \mathbb{I}_N) + B_{t-1}B^T, \quad \text{and hence,} \quad T^*_B H_t \cdot \Delta = \mathrm{mat}\left(B_t \cdot \mathrm{math}^*(\Delta)\right). \tag{3.16}$$

The computability constraints. In the particular case of the VEC(1,1) model, the existence of the matrix recursions (3.14)–(3.16) associated to (3.10)–(3.12) makes the computation of (3.8) relatively easy. For more general models, the matrix recursions are not easily available and one is forced to work directly with (the analog of) (3.10)–(3.12). In those cases, if we deal with a long time series sample, the computation of the gradient (3.8) may turn out numerically very expensive since it consists of the sum of T terms $T^*_\theta H_t \cdot \nabla_{H_t} l_t$, each of which is made of the sum of the t terms recursively defined in (3.10)–(3.12). A major simplification can be obtained if we restrict ourselves in the estimation process to matrices B whose top eigenvalue is in norm smaller than one. The defining expressions for the differential operators $T^*_\theta H_t$ show that in that situation, only a certain number of iterations, potentially small, is needed to compute the gradients with a prescribed precision. This is particularly visible in the expressions (3.5)–(3.7) where the dependence on the powers of B makes very small many of the involved summands whenever the spectrum of B is strictly contained in the unit disk. This is the reason why we will impose this as an additional estimation constraint. The details of this statement are spelled out in the proposition below that we present after the summary of the constraints that we will impose all along the paper on the model (3.1):

(SC) Stationarity constraints: $\mathbb{I}_N(1-\epsilon_{AB}) - (A+B)(A+B)^T \succeq 0$ for some small $\epsilon_{AB} > 0$.

(PC) Positivity constraints: $\mathrm{math}(\mathbf{c}) - \epsilon_{\mathbf{c}}\mathbb{I}_n \succeq 0$, $\Sigma(A) - \epsilon_A \mathbb{I}_{n^2} \succeq 0$, and $\Sigma(B) - \epsilon_B \mathbb{I}_{n^2} \succeq 0$, for some small $\epsilon_A, \epsilon_B, \epsilon_{\mathbf{c}} > 0$.

(CC) Computability constraints: $\mathbb{I}_N(1-\widetilde{\epsilon}_B) - BB^T \succeq 0$ for some small $\widetilde{\epsilon}_B > 0$.

Proposition 3.4 *Let $t \in \mathbb{N}$ be a fixed lag and let $T^*_\theta H_t$ be the differential operators defined by applying t times the recursions (3.10)-(3.12). Consider now the operators $T^*_\theta H^k_t$ obtained by truncating the recursions (3.10)-(3.12) after k iterations, $k < t$. If we assume that the coefficients $\mathbf{c}, A$, and B satisfy the constraints* **(SC)**, **(PC)**, *and* **(CC)** *then the error committed in the truncations can be estimated using the following inequalities satisfied by the operator norms:*

$$\|T^*_{\mathbf{c}} H_t - T^*_{\mathbf{c}} H^k_t\|_{\mathrm{op}} \leq \frac{2(1-\widetilde{\epsilon}_B)^k}{\widetilde{\epsilon}_B}, \tag{3.17}$$

$$\|E\left[T^*_A H_t - T^*_A H^k_t\right]\|_{\mathrm{op}} \leq \frac{2(1-\widetilde{\epsilon}_B)^k \|\mathbf{c}\|}{\epsilon_{AB}}, \tag{3.18}$$

$$\|E\left[T^*_B H_t - T^*_B H^k_t\right]\|_{\mathrm{op}} \leq \frac{2(1-\widetilde{\epsilon}_B)^k \|\mathbf{c}\|}{\epsilon_{AB}}. \tag{3.19}$$

Notice that the last two inequalities estimate the error committed in mean. As consequence of these relations, if we allow a maximum expected error δ in the computation of the gradient (3.8) then a lower bound for the number k of iterations that need to be carried out in (3.10)–(3.12) is:

$$k = \max\left\{ \frac{\log\left(\frac{\widetilde{\epsilon}_B \delta}{2}\right)}{\log(1-\widetilde{\epsilon}_B)}, \frac{\log\left(\frac{\widetilde{\epsilon}_B \epsilon_{AB} \delta}{2\epsilon_{\mathbf{c}}}\right)}{\log(1-\widetilde{\epsilon}_B)} \right\}. \tag{3.20}$$

Remark 3.5 The estimate (3.20) for the minimum number of iterations needed to reach a certain precision in the computation of the gradient is by no means sharp. Numerical experiments show that the figure produced by this formula is in general too conservative. Nevertheless, this expression is still very valuable for it explicitly shows the pertinence of the computability constraint **(CC)**.

Remark 3.6 We emphasize that the constraints **(SC)**, **(PC)**, and **(CC)** are sufficient conditions for stationarity, positivity, and computability, respectively, but by no means necessary. For example **(SC)** and **(CC)** could be replaced by the more economical (but also more restrictive) condition that imposes $A, B \in \mathbb{S}^+_N$ with $\lambda_{\max}(A+B) \leq (1-\epsilon_{AB})$. In this situation it can be easily shown that $\lambda_{\max}(B) < 1$ and hence the computability constrained is automatically satisfied.

4 Calibration via Bregman matrix divergences

In this section we present an efficient optimization method that, given a sample $\mathbf{z}$, provides the parameter value $\widehat{\boldsymbol{\theta}}$ corresponding to the VEC(1,1) model that fits it best by maximizing the quasi-loglikelihood (3.3) subjected to the constraints **(SC)**, **(PC)**, and **(CC)**. It can be proved under certain regularity hypotheses (see [Gou97, page 119]) that the quasi-loglikelihood estimator $\widehat{\boldsymbol{\theta}}$ is consistent and asymptotically normal:

$$\sqrt{T}(\widehat{\boldsymbol{\theta}} - \boldsymbol{\theta}_0) \xrightarrow[\text{dist}]{} N(0, \Omega_0) \quad \text{where} \quad \Omega_0 = A_0^{-1} B_0 A_0^{-1}, \quad \text{with} \tag{4.1}$$

$$A_0 = E_{\boldsymbol{\theta}_0}\left[-\frac{\partial^2 l_t(\boldsymbol{\theta}_0)}{\partial \boldsymbol{\theta} \partial \boldsymbol{\theta}^T}\right] \quad \text{and} \quad B_0 = E_{\boldsymbol{\theta}_0}\left[\frac{\partial l_t(\boldsymbol{\theta}_0)}{\partial \boldsymbol{\theta}} \frac{\partial l_t(\boldsymbol{\theta}_0)}{\partial \boldsymbol{\theta}^T}\right]. \tag{4.2}$$

These matrices are usually consistently estimated by replacing the expectations by their empirical means and the true value of the parameter $\boldsymbol{\theta}_0$ by the estimator $\widehat{\boldsymbol{\theta}}$:

$$\widehat{A}_0 = -\frac{1}{T}\sum_{i=1}^{T} \frac{\partial^2 l_t(\widehat{\boldsymbol{\theta}})}{\partial \boldsymbol{\theta} \partial \boldsymbol{\theta}^T}, \qquad \widehat{B}_0 = \frac{1}{T}\sum_{i=1}^{T} \frac{\partial l_t(\widehat{\boldsymbol{\theta}})}{\partial \boldsymbol{\theta}} \frac{\partial l_t(\widehat{\boldsymbol{\theta}})}{\partial \boldsymbol{\theta}^T}.$$

4.1 Constrained optimization via Bregman divergences

The optimization method that we will be carrying out to maximize the quasi-loglikelihood is based on the use of **Burg's matrix divergence**. This divergence is presented, for example, in [KSD09a] and it is a particular instance of a Bregman divergence. Bregman divergences are of much use in the context of machine learning (see for instance [DT07, KSD09b] and references therein).

In our situation we have opted for this technique as it allows for a particularly efficient treatment of the constraints in our problem, avoiding the need to solve additional secondary optimization problems. In order to make this more explicit it is worth mentioning that we also considered different approaches consisting of optimizing quadratically penalized local first or second order models with the positive semidefinite constraints **(PS)**, **(SC)**, and **(CC)**; since we were not able to find a closed form expression for the optimization step induced by this constrained local model, we were forced to use Lagrange duality. Even though the constraints admit a simple conic formulation that allowed us to explicitly formulate the problem, this approach finally resulted in a problem that is much more computationally demanding than just incorporating the constraints into the primal scheme using Bregman divergences, as we propose below.

Definition 4.1 *Let $X, Y \in \mathbb{S}_n$ and $\phi : \mathbb{S}_n \longrightarrow \mathbb{R}$ a strictly convex differentiable function. The* **Bregman matrix divergence** *associated to ϕ is defined by*

$$D_\phi(X, Y) := \phi(X) - \phi(Y) - \operatorname{trace}\left(\nabla\phi(Y)^T(X - Y)\right).$$

Bregman divergences are used to measure distance between matrices. Indeed, if we take the squared Frobenius norm as the function ϕ, that is $\phi(X) := \|X\|^2$, then $D_\phi(X, Y) := \|X - Y\|^2$. Other example is the **von Neumann divergence** which is the Bregman divergence associated to the entropy of the eigenvalues of a positive definite matrix; more explicitly, if X is a positive definite matrix with eigenvalues $\{\lambda_1, \ldots, \lambda_n\}$, then $\phi(X) := \sum_{i=1}^n (\lambda_i \log \lambda_i - \lambda_i)$. In our optimization problem we will be using **Burg's matrix divergence** (also called the **LogDet divergence** or **Stein's loss** in the statistics literature [JS61]) which is the Bregman divergence obtained out of the Burg entropy of the eigenvalues of a positive definite matrix, that is $\phi(X) := -\sum_{i=1}^n \log \lambda_i$, or equivalently $\phi(X) := -\log\det(X)$. The resulting Bregman divergence over positive definite matrices is

$$D_B(X, Y) := \operatorname{trace}(XY^{-1}) - \log\det(XY^{-1}) - n. \tag{4.3}$$

The three divergences that we just introduced are examples of **spectral** divergences, that is, the function ϕ that defines them can be written down as the composition $\phi = \varphi \circ \lambda$, where $\varphi : \mathbb{R}^n \longrightarrow \mathbb{R}$ is differentiable strictly convex function and $\lambda : \mathbb{S}_n \longrightarrow \mathbb{R}^n$ is the function that lists the eigenvalues of X in algebraically decreasing order. It can be seen (see Appendix A in [KSD09a]) that spectral Bregman matrix divergences are invariant by orthogonal conjugations, that is, for any orthogonal matrix $Q \in \mathbb{O}_n$:

$$D_\phi(Q^T X Q, Q^T Y Q) = D_\phi(X, Y).$$

Burg divergences are invariant by an even larger group since

$$D_B(M^T X M, M^T Y M) = D_B(X, Y),$$

for any square non-singular matrix M. Additionally, for any non-zero scalar α:

$$D_B(\alpha X, \alpha Y) = D_B(X, Y).$$

The use of Bregman divergences in matrix constrained optimization problems is substantiated by replacing the quadratic term in the local model, that generally uses the Frobenius distance, by a Bregman divergence that places the set outside the constraints at an infinite distance. More explicitly, suppose that the constraints of a optimization problem are formulated as a positive definiteness condition $A \succeq 0$ and that we want to find

$$\arg\min_{A \succeq 0} f(A),$$

by iteratively solving the optimization problems associated to penalized local models of the form

$$f_{A^{(n)}}(A) := f\left(A^{(n)}\right) + \left\langle \nabla f\left(A^{(n)}\right), A - A^{(n)} \right\rangle + \frac{L}{2} D_\phi(A, A^{(n)}). \tag{4.4}$$

If in this local model we take $\phi(X) := \|X\|^2$ and the elastic penalization constant L is small enough, the minimizer $\arg\min_{A \succeq 0} f_{A^{(n)}}(A)$ is likely to take place outside the constraints. However, if we use Burg's divergence D_B instead, and $A^{(n)}$ is positive definite, then so is $\arg\min_{A \succeq 0} f_{A^{(n)}}(A)$ for no matter what value of the parameter L. This is so because as A approaches the constraints, the term $D_\phi(A, A^{(n)})$ becomes increasingly close to infinity producing the effect that we just described; see Figure 4.1 for an illustration. The end result of using Bregman divergences is that *they reduce a constrained optimization problem to a series of local unconstrained ones.*

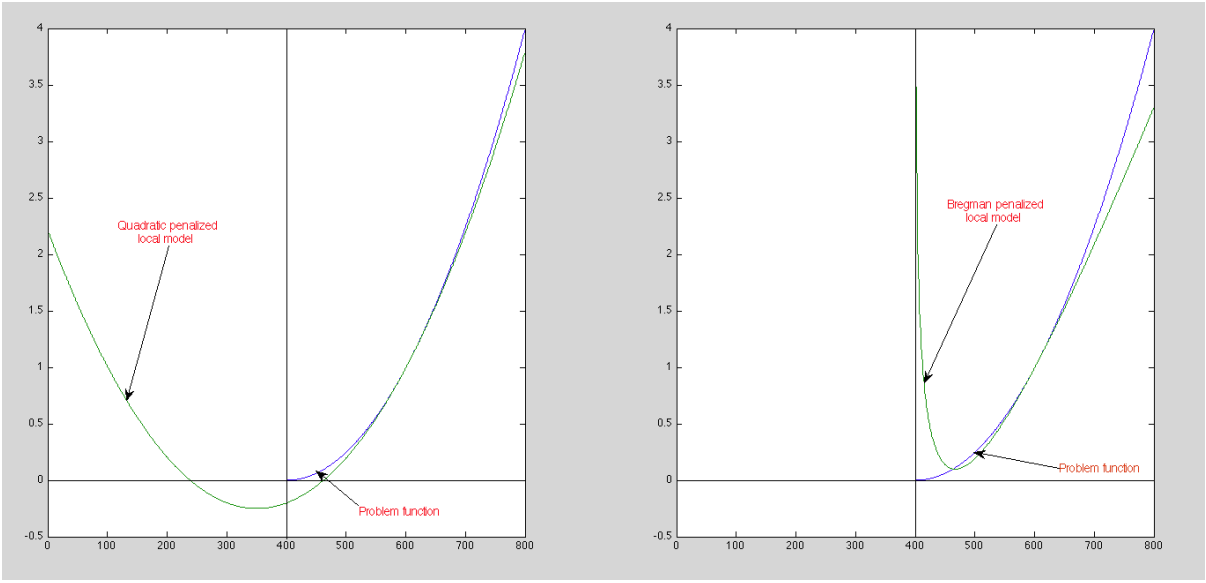

Figure 4.1: The blue function is subjected to the constraint $x \geq 400$ and, being strictly increasing, attains its minimum at $x = 400$. On the left hand side we use a standard quadratically penalized local model of the function and we see that its minimum is attained outside the constrained domain. On the right hand side we replace the quadratic penalization by a Bregman divergence that forces the local model to exhibit its optimum at a point that satisfies the constraints.

4.2 Bregman divergences for VEC models

Before we tackle the VEC estimation problem, we add to **(SC)**, **(PC)**, and **(CC)** a fourth constraint on the variable $\mathbf{c} \in \mathbb{R}^N$ that makes compact the optimization domain:

(KC) Compactness constraint: $K\mathbb{I}_N - \mathrm{math}(\mathbf{c}) \succeq 0$ for some $K \in \mathbb{R}$.

In practice the constant K is taken as a multiple of the Frobenius norm of the covariance matrix of the sample. This is a reasonable choice since by (3.2), in the stationary regime $\mathbf{c} = (\mathbb{I}_N - A - B)\mathrm{vech}(\Gamma(0))$; moreover, by the constraint (**SC**) and (2.5) we have

$$\|\mathbf{c}\| = \|(\mathbb{I}_N - A - B)\mathrm{vech}(\Gamma(0))\| \leq \|\mathbb{I}_N - A - B\|_{\mathrm{op}}\|\mathrm{vec}\|_{\mathrm{op}}\|\Gamma(0)\| \leq 2\|\Gamma(0)\|.$$

Now, given a sample $\mathbf{z}$ and a starting value for the parameters $\boldsymbol{\theta}_0 = (\mathbf{c}_0, A_0, B_0)$, our goal is finding the minimizer of minus the quasi-loglikelihood $f(\boldsymbol{\theta}) := -\log L(\mathbf{z}; \boldsymbol{\theta})$, subjected to the constraints **(SC)**, **(PC)**, **(CC)**, and **(KC)**. We will worry about the problem of finding a preliminary estimation $\boldsymbol{\theta}_0$ later on in Section 4.4. As we said before, our method is based on recursively optimizing penalized local models that incorporate Bregman divergences that ensure that the constraints are satisfied. More specifically, the estimate of the optimum $\boldsymbol{\theta}^{(n+1)}$ after n iterations is obtained by solving

$$\boldsymbol{\theta}^{(n+1)} = \underset{\boldsymbol{\theta} \in \mathbb{R}^N \times \mathbb{M}_N \times \mathbb{M}_N,}{\arg\min} \tilde{f}^{(n)}(\boldsymbol{\theta}), \tag{4.5}$$

where $\tilde{f}^{(n)}$ is defined by:

$$\begin{aligned}
\tilde{f}^{(n)}(\boldsymbol{\theta}) &= f(\boldsymbol{\theta}^{(n)}) + \langle \nabla f(\boldsymbol{\theta}^{(n)}), \boldsymbol{\theta} - \boldsymbol{\theta}^{(n)} \rangle + \frac{L_1}{2} D_B(\mathbb{I}_N - (A+B)^T(A+B), \mathbb{I}_N - (A^{(n)} + B^{(n)})^T(A^{(n)} + B^{(n)})) \\
&+ \frac{L_2}{2} D_B(\Sigma(A), \Sigma(A^{(n)})) + \frac{L_3}{2} D_B(\Sigma(B), \Sigma(B^{(n)})) + \frac{L_4}{2} D_B(\mathbb{I}_N - B^T B, \mathbb{I}_N - B^{(n)\,T} B^{(n)}) \\
&+ \frac{L_5}{2} D_B(\mathrm{math}(\mathbf{c}), \mathrm{math}(\mathbf{c}^{(n)})) + \frac{L_6}{2} D_B(K\mathbb{I}_N - \mathrm{math}(\mathbf{c}), K\mathbb{I}_N - \mathrm{math}(\mathbf{c}^{(n)})).
\end{aligned} \tag{4.6}$$

Notice that for the sake of simplicity we have incorporated the constraints in the divergences with the constraint tolerances $\epsilon_{AB}, \epsilon_A, \epsilon_B, \widetilde{\epsilon}_B$, and $\epsilon_{\mathbf{c}}$ set equal to zero.

The local optimization problem in (4.5) is solved by finding the value $\boldsymbol{\theta}_0$ for which

$$\nabla \tilde{f}^{(n)}(\boldsymbol{\theta}_0) = 0. \tag{4.7}$$

A long but straightforward computation shows that the gradient $\nabla \tilde{f}^{(n)}(\boldsymbol{\theta})$ is given by the expressions:

$$\begin{aligned}
\nabla_A \tilde{f}^{(n)}(\boldsymbol{\theta}) &= \nabla_A f(\boldsymbol{\theta}^{(n)}) - L_1(A+B)\left(\left(\mathbb{I}_N - (A^{(n)} + B^{(n)})^T(A^{(n)} + B^{(n)})\right)^{-1} - \left(\mathbb{I}_N - (A+B)^T(A+B)\right)^{-1}\right) \\
&\quad + \frac{L_2}{2}\Sigma^*\left(\Sigma(A^{(n)})^{-1} - \Sigma(A)^{-1}\right),
\end{aligned} \tag{4.8}$$

$$\begin{aligned}
\nabla_B \tilde{f}^{(n)}(\boldsymbol{\theta}) &= \nabla_B f(\boldsymbol{\theta}^{(n)}) - L_1(A+B)\left(\left(\mathbb{I}_N - (A^{(n)} + B^{(n)})^T(A^{(n)} + B^{(n)})\right)^{-1} - \left(\mathbb{I}_N - (A+B)^T(A+B)\right)^{-1}\right) \\
&\quad + \frac{L_3}{2}\Sigma^*\left(\Sigma(B^{(n)})^{-1} - \Sigma(B)^{-1}\right) - L_4 B\left(\left(\mathbb{I}_N - B^{(n)\,T}B^{(n)}\right)^{-1} - \left(\mathbb{I}_N - B^T B\right)^{-1}\right),
\end{aligned} \tag{4.9}$$

$$\begin{aligned}
\nabla_{\mathbf{c}} \tilde{f}^{(n)}(\boldsymbol{\theta}) &= \nabla_{\mathbf{c}} f(\boldsymbol{\theta}^{(n)}) + \frac{L_5}{2}\mathrm{math}^*\left(\mathrm{math}(\mathbf{c}^{(n)})^{-1} - \mathrm{math}(\mathbf{c})^{-1}\right) \\
&\quad - \frac{L_6}{2}\mathrm{math}^*\left((K\mathbb{I}_N - \mathrm{math}(\mathbf{c}^{(n)}))^{-1} - (K\mathbb{I}_N - \mathrm{math}(\mathbf{c}))^{-1}\right),
\end{aligned} \tag{4.10}$$

where $\nabla_{\boldsymbol{\theta}} f(\boldsymbol{\theta}^{(n)}) = -\nabla_{\boldsymbol{\theta}} \log L(\mathbf{z};\boldsymbol{\theta}^{(n)})$ is provided by the expressions in Proposition 3.3. We will numerically find the solution of the equation (4.7) using the Newton-Raphson algorithm, which requires computing the tangent map to $\nabla \tilde{f}^{(n)}(\boldsymbol{\theta})$. In order to do so, let $g_1^{(n)}(A,B)$, $g_2^{(n)}(A,B)$, and $g_3^{(n)}(\mathbf{c})$ be the functions in the right hand side of the expressions (4.8), (4.9), and (4.10), respectively, and $g^{(n)}(A,B,\mathbf{c}) := \left(g_1^{(n)}(A,B), g_2^{(n)}(A,B), g_3^{(n)}(\mathbf{c})\right)$; additionally, define the map $\Lambda(A) : \mathbb{M}_N \to \mathbb{M}_N$ by $\Lambda(A) := \mathbb{I}_N - A^T A$, as well as

$$\Xi_A^{(n)}(\Delta) = -\Delta\left(\Lambda\left(A^{(n)}\right)^{-1} - \Lambda(A)^{-1}\right) + A\Lambda(A)^{-1}\left(\Delta^T(A) + A^T\Delta\right)\Lambda(A)^{-1}, \tag{4.11}$$

$$\mathfrak{X}_A(\Delta) = \Sigma^*\left(\Sigma(A)^{-1}\Sigma(\Delta)\Sigma(A)^{-1}\right), \tag{4.12}$$

for any $A, \Delta \in \mathbb{M}_N$. A straightforward computation shows that:

$$T_{(A,B)}g_1^{(n)}\cdot(\Delta_A,\Delta_B) = \left(L_1\Xi_{A+B}^{(n)}(\Delta_A) + \frac{L_2}{2}\mathfrak{X}_A(\Delta_A), L_1\Xi_{A+B}^{(n)}(\Delta_B)\right), \tag{4.13}$$

$$T_{(A,B)}g_2^{(n)}\cdot(\Delta_A,\Delta_B) = \left(L_1\Xi_{A+B}^{(n)}(\Delta_A), L_1\Xi_{A+B}^{(n)}(\Delta_B) + \frac{L_3}{2}\mathfrak{X}_B(\Delta_B) + L_4\Xi_B^{(n)}(\Delta_B)\right), \tag{4.14}$$

$$\begin{aligned} T_{\mathbf{c}}g_3^{(n)}\cdot\Delta_{\mathbf{c}} &= \frac{L_5}{2}\mathrm{math}^*\left(\mathrm{math}(\mathbf{c})^{-1}\mathrm{math}(\Delta_{\mathbf{c}})\mathrm{math}(\mathbf{c})^{-1}\right) \\ &\quad + \frac{L_6}{2}\mathrm{math}^*\left((K\mathbb{I}_N - \mathrm{math}(\mathbf{c}))^{-1}\mathrm{math}(\Delta_{\mathbf{c}})(K\mathbb{I}_N - \mathrm{math}(\mathbf{c}))^{-1}\right) \end{aligned} \tag{4.15}$$

In obtaining these equalities we used that the tangent map to the matrix inversion operation $\mathrm{inv}(X) := X^{-1}$ is given by $T_X\mathrm{inv}\cdot\Delta = -X^{-1}\Delta X^{-1}$ and hence

$$T_A\Lambda(A)^{-1}\cdot\Delta_A = \Lambda(A)^{-1}\left(\Delta_A^T A + A^T\Delta_A\right)\Lambda(A)^{-1} \quad \text{and} \quad T_A\Sigma(A)^{-1}\cdot\Delta_A = -\Sigma(A)^{-1}\Sigma(\Delta_A)\Sigma(A)^{-1}.$$

The use of the tangent maps (4.13)–(4.15) in a numerical routine that implements the Newton-Raphson method requires computing the matrix associated to the linear map $T_{(A,B,\mathbf{c})}g^{(n)}$. A major part in this task, namely the matrix associated to the map $\Xi^{(n)}$ in (4.11), admits a closed form expression that avoids a componentwise computation. Indeed:

$$\begin{aligned} \mathrm{vec}(\Xi_A^{(n)}(\Delta)) &= -\left[\left(\Lambda\left(A^{(n)}\right)^{-1} - \Lambda(A)^{-1}\right)\otimes\mathbb{I}_N\right]\mathrm{vec}(\Delta) + \left[\left(A\Lambda(A)^{-1}\right)^T\otimes A\Lambda(A)^{-1}\right]K_{NN}\mathrm{vec}(\Delta) \\ &\quad + \left[\Lambda(A)^{-1\,T}\otimes A\Lambda(A)^{-1}A^T\right]\mathrm{vec}(\Delta), \end{aligned} \tag{4.16}$$

where K_{NN} is the (N,N)-commutation matrix (see [MN79]). This expression implies that the matrix $\widetilde{\Xi_A^{(n)}} \in \mathbb{M}_{N^2}$ associated to the linear map $\Xi_A^{(n)} : \mathbb{M}_N \to \mathbb{M}_N$ is given by

$$\widetilde{\Xi_A^{(n)}} = -\left[\left(\Lambda\left(A^{(n)}\right)^{-1} - \Lambda(A)^{-1}\right)\otimes\mathbb{I}_N\right] + \left[\left(A\Lambda(A)^{-1}\right)^T\otimes A\Lambda(A)^{-1}\right]K_{NN} + \left[\Lambda(A)^{-1\,T}\otimes A\Lambda(A)^{-1}A^T\right]. \tag{4.17}$$

In order to obtain (4.16), we used the following properties of the vec operator:

$$\mathrm{vec}(AB) = \left(B^T\otimes\mathbb{I}\right)\mathrm{vec}(A), \quad \mathrm{vec}(ABC) = \left(C^T\otimes A\right)\mathrm{vec}(A) \quad \text{and} \quad \mathrm{vec}(A^T) = K_{NN}\mathrm{vec}(A),$$

for any $A, B, C \in \mathbb{M}_N$. We have not found a closed formula for the matrices associated to the other linear maps that constitute (4.13)–(4.15) and hence they need to be obtained in a componentwise manner by

applying them to all the elements of a canonical basis. Let $\widetilde{\mathfrak{X}_A}$, $\widetilde{T_{(A,B,\mathbf{c})}g^{(n)}}$, and $\widetilde{T_{\mathbf{c}}g_3^{(n)}}$ be the matrices associated to $\mathfrak{X}_A$, $T_{(A,B,\mathbf{c})}g^{(n)}$, and $T_{\mathbf{c}}g_3^{(n)}$, respectively. Then, by (4.13)–(4.15), we have:

$$\widetilde{T_{(A,B,\mathbf{c})}g^{(n)}} = \begin{pmatrix} L_1\widetilde{\Xi^{(n)}_{A+B}} + \frac{L_2}{2}\widetilde{\mathfrak{X}_A} & L_1\widetilde{\Xi^{(n)}_{A+B}} & \mathbf{0} \\ L_1\widetilde{\Xi^{(n)}_{A+B}} & L_1\widetilde{\Xi^{(n)}_{A+B}} + \frac{L_3}{2}\widetilde{\mathfrak{X}_B} + L_4\widetilde{\Xi^{(n)}_{B}} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \widetilde{T_{\mathbf{c}}g_3^{(n)}} \end{pmatrix}. \tag{4.18}$$

Using this matrix, the solution $\boldsymbol{\theta}_0$ of (4.7) is the limit of the sequence $\{\boldsymbol{\theta}^{(n,\,k)}\}_{k\in\mathbb{N}} := \{(A^{(n,\,k)}, B^{(n,\,k)}, \mathbf{c}^{(n,\,k)})\}_{k\in\mathbb{N}}$ constructed using the prescription $\boldsymbol{\theta}^{(n,\,1)} = \boldsymbol{\theta}^{(n)} = (A^{(n)}, B^{(n)}, \mathbf{c}^{(n)})$ and by iteratively solving the linear systems:

$$\widetilde{T_{\boldsymbol{\theta}^{(n,\,k)}}g^{(n)}} \cdot \begin{pmatrix} \mathrm{vec}(A^{(n,\,k+1)}) \\ \mathrm{vec}(B^{(n,\,k+1)}) \\ \mathbf{c}^{(n,\,k+1)} \end{pmatrix} = -\mathrm{vec}\left(\nabla\tilde{f}^{(n)}(\boldsymbol{\theta}^{(n,\,k)})\right) + \widetilde{T_{\boldsymbol{\theta}^{(n,\,k)}}g^{(n)}} \cdot \begin{pmatrix} \mathrm{vec}(A^{(n,\,k)}) \\ \mathrm{vec}(B^{(n,\,k)}) \\ \mathbf{c}^{(n,\,k)} \end{pmatrix}. \tag{4.19}$$

Remark 4.2 Since the tangent map $T_{\boldsymbol{\theta}}g^{(n)}$ can be assimilated to the Hessian of $\tilde{f}^{(n)}$, its matricial expression $\widetilde{T_{\boldsymbol{\theta}}g^{(n)}}$ in (4.18) should be symmetric. When this matrix is actually numerically constructed, the part resulting from the matrix identity (4.17) is automatically symmetric. The rest, that comes out of a componentwise study, may introduce numerical differences that slightly spoil symmetricity and that, in practice, has a negative effect in the performance of the optimization algorithm as a whole. That is why we strongly advice to symmetrize by hand $\widetilde{T_{\boldsymbol{\theta}}g^{(n)}}$ once it has been computed.

4.3 Performance improvement: BFGS and trust-region corrections

The speed of convergence of the estimation algorithm presented in the previous section can be significantly increased by enriching the local model with a quadratic BFGS (Broyden-Fletcher-Goldfarb-Shanno) type term and by only accepting steps of a certain quality measured by the ratio between the actual descent and that predicted by the local model (see [CGT00] and references therein).

The BFGS correction is introduced by adding to the local penalized model $\tilde{f}^{(n)}(\boldsymbol{\theta})$ defined in (4.6), the BFGS Hessian proxy $H^{(n)}$ iteratively defined by:

$$H^{(n)} = H^{(n-1)} + \frac{y^{(n-1)}y^{(n-1)\,T}}{y^{(n-1)\,T}s^{(n-1)}} - \frac{H^{(n-1)}s^{(n-1)}s^{(n-1)\,T}H^{(n-1)}}{s^{(n-1)\,T}H^{(n-1)}s^{(n-1)}}.$$

with $H^{(0)}$ an arbitrary positive semidefinite matrix and where $s^{(n-1)} := \boldsymbol{\theta}^{(n)} - \boldsymbol{\theta}^{(n-1)}$ and $y^{(n-1)} := \nabla f(\boldsymbol{\theta}^{(n)}) - \nabla f(\boldsymbol{\theta}^{(n-1)})$. More specifically, we replace the local penalized model $\tilde{f}^{(n)}(\boldsymbol{\theta})$ by

$$\hat{f}^{(n)}(\boldsymbol{\theta}) := \tilde{f}^{(n)}(\boldsymbol{\theta}) + \frac{1}{2}\left(\boldsymbol{\theta} - \boldsymbol{\theta}^{(n)}\right)^T H^{(n)}\left(\boldsymbol{\theta} - \boldsymbol{\theta}^{(n)}\right),$$

whose gradient is obviously given by:

$$\hat{g}^{(n)}(\boldsymbol{\theta}) := \nabla\hat{f}^{(n)}(\boldsymbol{\theta}) = \nabla\tilde{f}^{(n)}(\boldsymbol{\theta}) + H^{(n)}\left(\boldsymbol{\theta} - \boldsymbol{\theta}^{(n)}\right) = \widetilde{g}^{(n)}(\boldsymbol{\theta}) + H^{(n)}\left(\boldsymbol{\theta} - \boldsymbol{\theta}^{(n)}\right),$$

with $\widetilde{g}^{(n)}(\boldsymbol{\theta}) = \nabla\tilde{f}^{(n)}(\boldsymbol{\theta})$ given by (4.8)–(4.10). Using this corrected local penalized model, the solution of the optimization problem will be obtained by iteratively computing

$$\boldsymbol{\theta}^{(n+1)} = \underset{\boldsymbol{\theta}\in\mathbb{R}^N\times\mathbb{M}_N\times\mathbb{M}_N,}{\arg\min}\ \hat{f}^{(n)}(\boldsymbol{\theta}). \tag{4.20}$$

This is carried out by finding the solution $\boldsymbol{\theta}_0$ of the equation

$$\hat{g}^{(n)}(\boldsymbol{\theta}_0) = \widetilde{g}^{(n)}(\boldsymbol{\theta}_0) + H^{(n)}\left(\boldsymbol{\theta}_0 - \boldsymbol{\theta}^{(n)}\right) = 0. \tag{4.21}$$

using a modified version of the Newton-Raphson iterative scheme spelled out in (4.19). Indeed, it is easy to show that $\boldsymbol{\theta}_0$ is the limit of the sequence $\{\boldsymbol{\theta}^{(n,\,k)}\}_{k\in\mathbb{N}}$ constructed exactly as in Section 4.2 where the linear systems (4.19) are replaced by

$$\left(\widetilde{T_{\boldsymbol{\theta}^{(n,\,k)}} g^{(n)}} + \widetilde{H^{(n)}}\right) \cdot \widetilde{\boldsymbol{\theta}^{(n,\,k+1)}} = -\mathrm{vec}\left(\nabla \tilde{f}^{(n)}(\boldsymbol{\theta}^{(n,\,k)})\right) + \widetilde{H^{(n)}} \cdot \widetilde{\boldsymbol{\theta}^{(n)}} + \widetilde{T_{\boldsymbol{\theta}^{(n,\,k)}} g^{(n)}} \cdot \widetilde{\boldsymbol{\theta}^{(n,\,k)}}. \tag{4.22}$$

where $\widetilde{\boldsymbol{\theta}^{(n,\,k+1)}} = \begin{pmatrix} \mathrm{vec}(A^{(n,\,k+1)}) \\ \mathrm{vec}(B^{(n,\,k+1)}) \\ \mathbf{c}^{(n,\,k+1)} \end{pmatrix}$ and $\widetilde{H^{(n)}} \in \mathbb{M}_{2N^2+N}$ denotes the matrix associated to $H^{(n)}$ that satisfies

$$\mathrm{vec}\left(H^{(n)} \cdot \boldsymbol{\theta}\right) = \widetilde{H^{(n)}} \cdot \begin{pmatrix} \mathrm{vec}(A) \\ \mathrm{vec}(B) \\ \mathbf{c} \end{pmatrix} \quad \text{for any} \quad \boldsymbol{\theta} = (A, B, \mathbf{c}).$$

Important remark: the Newton-Raphson method and the constraints. In Section 4.1 we explained how the use of Bregman divergences ensures that at each iteration, the extremum of the local penalized model satisfies the constraints of the problem. However, the implementation of the Newton-Raphson method that provides the root of the equation (4.21) does not, in general, respect the constraints, and hence this point requires special care.

In the construction of our optimization algorithm we have used the following prescription in order to ensure that all the elements of the sequence $\{\boldsymbol{\theta}^{(n,\,k)}\}_{k\in\mathbb{N}}$ that converge to the root $\boldsymbol{\theta}_0$ satisfy the constraints: given $\boldsymbol{\theta}^{(n,\,1)} = \boldsymbol{\theta}^{(n)}$ (that satisfies the constraints) let $\boldsymbol{\theta}^{(n,\,2)}$ be the second value in the Newton-Raphson sequence obtained by solving the linear system (4.22). If the value $\boldsymbol{\theta}^{(n,\,2)}$ thereby constructed satisfies the constraints it is then accepted and we continue to the next iteration; otherwise we set

$$\boldsymbol{\theta}^{(n,\,2)} := \boldsymbol{\theta}^{(n,\,1)} + \frac{\boldsymbol{\theta}^{(n,\,2)} - \boldsymbol{\theta}^{(n,\,1)}}{2} \tag{4.23}$$

iteratively until $\boldsymbol{\theta}^{(n,\,2)}$ satisfies the constraints. Notice that by repeatedly performing (4.23), the value $\boldsymbol{\theta}^{(n,\,2)}$ hence constructed is closer and closer to $\boldsymbol{\theta}^{(n,\,1)}$; since this latter point satisfies the constraints, so will at some point $\boldsymbol{\theta}^{(n,\,2)}$. This manipulation that took us from $\boldsymbol{\theta}^{(n,\,1)}$ to $\boldsymbol{\theta}^{(n,\,2)}$ in a constraint compliant fashion has to be carried out at each iteration to go from $\boldsymbol{\theta}^{(n,\,k)}$ to $\boldsymbol{\theta}^{(n,\,k+1)}$.

Trust-region iteration acceptance correction: given an starting point $\boldsymbol{\theta}^0$ we have given a prescription for the construction of a sequence $\{\boldsymbol{\theta}^{(n)}\}_{n\in\mathbb{N}}$ that converges to the constrained minimizer of minus the quasi-loglikelihood $f(\boldsymbol{\theta}) := -\log L(\mathbf{z};\boldsymbol{\theta})$. We now couple this optimization routine with a trust-region technique. The trust-region algorithm provides us with a systematic method to test the pertinence of an iteration before it is accepted and to adaptively modify the strength of the local penalization in order to speed up the convergence speed. In order to carefully explain our use of this procedure consider first the local model (4.5) in which all the constants $L_1, \ldots, L_6$ that manage the strength of the constraint penalizations are set to a common value L. At each iteration of (4.20) compute the **adequacy ratio** $\rho^{(n)}$ defined as

$$\rho^{(n)} := \frac{f(\boldsymbol{\theta}^{(n)}) - f(\boldsymbol{\theta}^{(n-1)})}{\hat{f}^{(n)}(\boldsymbol{\theta}^{(n)}) - \hat{f}^{(n)}(\boldsymbol{\theta}^{(n-1)})} \tag{4.24}$$

which measures how close the descent in the target function in the present iteration is to the one exhibited by the local model $\hat{f}^{(n)}$. The values that can be obtained for $\rho^{(n)}$ are classified into three categories that determine different courses of action:

1. **Too large step** $\rho^{(n)} < 0.01$: there is too much dissimilarity between the local penalized model and the actual target function. In this situation, the iteration update is rejected by setting $\boldsymbol{\theta}^{(n)} = \boldsymbol{\theta}^{(n-1)}$ and the penalization is strengthened by doubling the constant: $L = 2L$

2. **Good step** $0.01 \leq \rho^{(n)} \leq 0.9$: the iteration update is accepted and the constant L is left unchanged.

3. **Too small step** $0.9 \leq \rho^{(n)}$: the iteration update is accepted but given the very good adequacy between the local penalized model and the target function we can afford loosening the penalization strength by setting $L = \frac{1}{2}L$ as the constant that will be used in the next iteration.

Remark 4.3 Even though the definition of the adequacy ratio in (4.24) uses the full penalized local models $\hat{f}^{(n)}$, we have seen that in practice the linear approximation suffices to obtain good results.

4.4 Preliminary estimation

As any optimization algorithm, the one that we just presented requires a starting point $\boldsymbol{\theta}^{(0)}$. The choice of a good preliminary estimation of $\boldsymbol{\theta}^{(0)}$ is particularly relevant in our situation since the quasi-loglikelihood exhibits generically local extrema and hence initializing the optimization algorithm close enough to the solution may prove to be crucial in order to obtain the correct solution.

Given a sample $\mathbf{z} = \{\mathbf{z}_1, \ldots, \mathbf{z}_T\}$, a reasonable estimation for $\boldsymbol{\theta}^{(0)}$ can be obtained by using the following two steps scheme:

1. Find a preliminary estimation of the conditional covariance matrices sequence $\{H_1, \ldots, H_T\}$ out of the sample $\mathbf{z}$. This can be achieved by using a variety of existing non-computationally intensive techniques. A non-exhaustive list is:

(i) Orthogonal GARCH model (O-GARCH): introduced in [Din94, AC97, Ale98, Ale03]; this technique is based on fitting one-dimensional GARCH models to the principal components obtained out of the sample marginal covariance matrix of $\mathbf{z}$.

(ii) Generalized orthogonal GARCH model (GO-GARCH) [vdW02]: similar to O-GARCH, but in this case the one-dimensional modeling is carried out not for the principal components of $\mathbf{z}$ but for its image with respect to a transformation V which is assumed is assumed to be just invertible (in the case of O-GARCH is also orthogonal) and it is estimated directly via a maximum likelihood procedure, together with the parameters of the one-dimensional GARCH models. GO-GARCH produces better empirical results than O-GARCH but it lacks the factoring estimation feature that O-GARCH has, making it more complicated for the modeling of large dimensional time series and conditional covariance matrices.

(iii) Independent component analysis (ICA-GARCH): [WYL06, GFGPP08] this model is based on a signal separation technique [Com94, HO97] that turns the time series into statistically independent components that are then treated separately using one dimensional GARCH models.

(iv) Dynamic conditional correlation model (DCC): introduced in [TT02, Eng02], this model proposes a dynamic behavior of the conditional correlation that depends on a small number of parameters and that nevertheless is still capable of capturing some of the features of more complicated multivariate models. Moreover, a version of this model [ES01] can be estimated consistently using a two-step approach that makes it suitable to handle large dimensional problems.

Another method that is widely used in the context of financial log-returns is the one advocated by Riskmetrics [Ris96] that proposes exponentially weighted moving average (EWMA) models for the time

evolution of variances and covariances; this comes down to working with IGARCH type models with a coefficient that is not estimated but proposed by Riskmetrics and that is the same for all the factors.

2. Estimation of $\boldsymbol{\theta}^{(0)}$ out of $\mathbf{z}$ and $H = \{H_t\}_{t\in\{1,\dots,T\}}$ using constrained ordinary least squares. If we have the sample $\mathbf{z}$ and a preliminary estimation of the conditional covariances $\{H_t\}_{t\in\{1,\dots,T\}}$, a good candidate for $\boldsymbol{\theta}^{(0)} = (A^{(0)}, B^{(0)}, \mathbf{c}^{(0)})$ is the value that minimizes the sum of the Euclidean norms $s_t := \|\mathbf{h}_t - \left(\mathbf{c} + A\boldsymbol{\eta}_{t-1} + B\mathbf{h}_{t-1}\right)\|^2$, that is,

$$s(A, B, \mathbf{c}; \mathbf{z}, H) = \sum_{t=2}^{T} s_t(A, B, \mathbf{c}; \mathbf{z}, H) = \sum_{t=2}^{T} \|\mathbf{h}_t - \left(\mathbf{c} + A\boldsymbol{\eta}_{t-1} + B\mathbf{h}_{t-1}\right)\|^2,$$

subjected to the constraints **(SC)**, **(PC)**, **(CC)**, and **(KC)**. This minimizer can be efficiently found by using the Bregman divergences based method introduced in Sections 4.1 through 4.3 with the function $s(A, B, \mathbf{c}; \mathbf{z}, H)$ replacing minus the log-likelihood. However, we emphasize that unlike the situation in the log-likelihood problem, the choice of a starting point in the optimization of $s(A, B, \mathbf{c}; \mathbf{z}, H)$ is irrelevant given the convexity of his function.

As a consequence of these arguments, the preliminary estimation $\boldsymbol{\theta}^{(0)}$ is obtained by iterating (4.20) where in the local model (4.6) the map f is replaced by s. This scheme is hence readily applicable once the gradient of s, provided by the following formulas, is available:

$$\begin{aligned}
\nabla_A s &= 2\sum_{t=2}^{T} \left[A\boldsymbol{\eta}_{t-1}\boldsymbol{\eta}_{t-1}^T + \mathbf{c}\boldsymbol{\eta}_{t-1}^T + B\mathbf{h}_{t-1}\boldsymbol{\eta}_{t-1}^T - \mathbf{h}_t\boldsymbol{\eta}_{t-1}^T\right], \\
\nabla_B s &= 2\sum_{t=2}^{T} \left[\mathbf{c}\mathbf{h}_{t-1}^T + A\boldsymbol{\eta}_{t-1}\mathbf{h}_{t-1}^T + B\mathbf{h}_{t-1}\mathbf{h}_{t-1}^T - \mathbf{h}_t\mathbf{h}_{t-1}^T\right], \\
\nabla_{\mathbf{c}} s &= 2\sum_{t=2}^{T} \left[\mathbf{c} + A\boldsymbol{\eta}_{t-1} + B\mathbf{h}_{t-1} - \mathbf{h}_t\right].
\end{aligned}$$

5 Numerical experiments

In this section we illustrate the estimation method presented in Section 4 with various simulations that give an idea of the associated computational effort and of the pertinence of the VEC model in different dimensions.

The data set. We have used in our experiments the daily closing prices between January 3, 2005 and December 31, 2009 (that is, 1258 date entries) of the stock associated to the companies Alcoa, Apple, Abbott Laboratories, American Electric, Allstate, Amgen, Amazon.com, and Avon. All these stocks are traded at the NYSE in US dollars and, in the last date of our sample, they were all constituents of the S&P500 index. The quotes are adjusted with respect to dividend payments and stock splits. Figure 5.1 represents graphically the data set.

Computational effort associated to the estimation method. In table 5.1 we have gathered the required computing time and the necessary gradient calls to fit VEC(1,1) models to the log-returns of our data set in different dimensions. In the $n = 1$ column we present the results associated to fitting a VEC model to the log-returns of the first element of the data set; the same in the $n = 2$ column with respect to the log-returns of the first two elements of the data set, and so on. The stopping criterion for the algorithm is established by setting a termination tolerance on the function value equal to 10^{-5}. The last row of the table shows how the algorithm becomes increasingly costlier with the dimensionality of the problem when the BFGS correction is dropped.The results of this experiment suggest that the

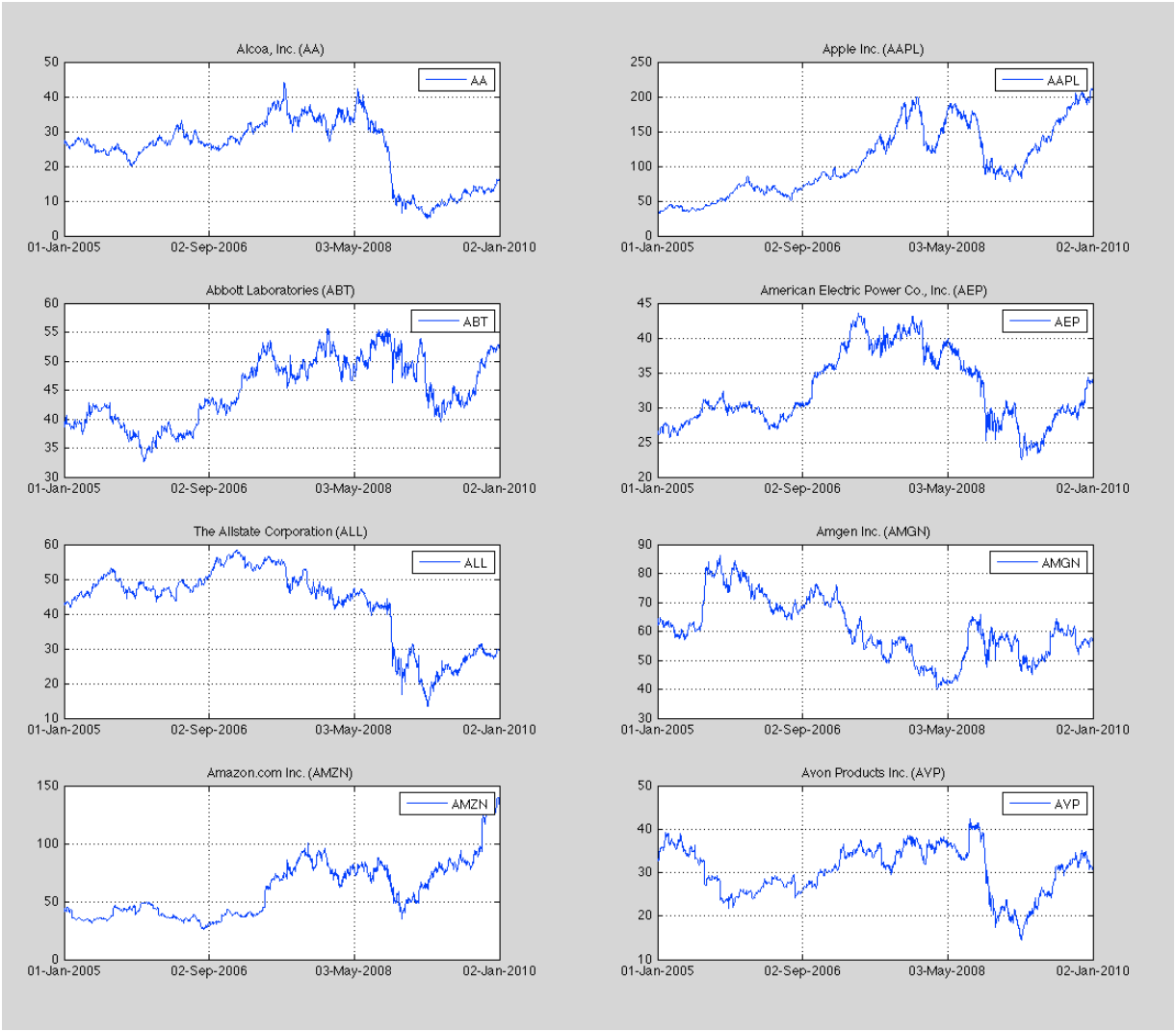

Figure 5.1: Stock quotes used in the numerical experiments. The quotes represent closing prices adjusted with respect to dividend payments and stock splits. Source: Yahoo Finance.

trust-region correction speeds up the algorithm and the BFGS modification makes the convergence rate dimensionally independent.

	Computation time and gradient calls											
	$n = 1$ (3 parameters)		$n = 2$ (21 parameters)		$n = 3$ (78 parameters)		$n = 4$ (210 parameters)		$n = 5$ (465 parameters)		$n = 6$ (903 parameters)	
	Grad. calls	Time	Grad. calls	Time	Grad. calls	Time	Grad. calls	Time	Grad. calls	Time	Grad. calls	Time
Full method	50	1.45 sec	97	58 sec	99	3 min 9 sec	94	18 min	85	73 min	105	5 hrs 36 min
No BFGS	106	2.30 sec	281	159 sec	378	8 min 57 sec	404	47 min	534	298 min	591	22 hr

Table 5.1: Computation time and gradient calls required when running the estimation method presented in Section 4, with and without the BFGS correction. The simulations were carried out using a nonparallelized Matlab script on an Apple computer endowed with two double core 3 GHz processors, 64 bits.

Variance minimizing portfolios, proxy replication, and spectral sparsity. As we have already pointed out several times, the main concern when using VEC models lays in the overabundance of parameters, whose number may easily be bigger than the sample size in standard applications, even when dealing with low dimensional problems. This lack of parsimony already appears when dealing with our data set for it contains 1257 historical log-returns, while the VEC(1,1) model requires 1596 parameters in dimension seven and 2628 in dimension 8.

The goal of the following experiment consists of assessing how serious this problem is. More explicitly, we will study how the pertinence of VEC as a modeling tool evolves with the increase in dimensionality, when compared with other more parsimonious and widely used alternatives, namely:

- Exponentially Weighted Moving Average (EWMA) model for the conditional covariance matrices with the autroregressive coefficient $\lambda = 0.94$ proposed by Riskmetrics [Ris96] for daily data.
- Orthogonal GARCH model (OGARCH), as in [Din94, AC97, Ale98, Ale03].
- Dynamic conditional correlation model (DCC) of [TT02, Eng02].

These modeling approaches will be tested by evaluating:

- **Comparative performance in the construction of dynamic variance minimizing portfolios:** all the models that we just enumerated and that take part in our comparison share the form
$$\mathbf{z}_t = H_t^{1/2}\boldsymbol{\epsilon_t} \qquad \text{with} \qquad \{\boldsymbol{\epsilon_t}\} \sim \mathrm{IIDN}(\mathbf{0}, \boldsymbol{I}_n), \tag{5.1}$$
where $\{H_t\}$ is a predictable matrix process. What changes from model to model is the specification that determines the dynamical behavior of $\{H_t\}$; in the particular case of VEC(1,1), that specification is spelled out in (3.1). When (5.1) is fitted to the log-returns associated to our data set, the matrices $\{H_t\}$ provide an (model dependent) estimate of the conditional covariance of the log-returns process. Moreover, it is not difficult to show that if $\mathbf{w} = (w_1, \ldots, w_n)'$ is a weights vector such that $\sum_{i=1}^n w_i = 1$, then the conditional variance of the net returns process of the associated portfolio is given by $\{\mathbf{w}^T A_t \mathbf{w}\}$, where A_t is the matrix whose (i,j) entry A_{ij}^t is given by
$$A_{ij}^t = \exp\left(\sum_{k=1}^n h_{ik}h_{jk}\right) - 1,$$
with h_{ij}^t the (i,j) entry of the matrix H_t. A dynamic variance minimizing portfolio is a weights vector $\mathbf{w}_t$ defined as the solution at each time step of the optimization problem
$$\underset{\mathbf{w}\in\mathbb{R}^n,\, \sum_{i=1}^n w_i = 1}{\arg\min} \quad \mathbf{w}^T A_t \mathbf{w}. \tag{5.2}$$

A straightforward application of Lagrange duality shows that the solutions $\mathbf{w}_t$ of (5.2) are given by either the zero eigenvectors of A_t, or by

$$\mathbf{w}_t = \frac{1}{\mathbf{i}^T A_t^{-1} \mathbf{i}} A_t^{-1} \mathbf{i}, \tag{5.3}$$

when A_t is invertible, where $\mathbf{i}$ is an n-dimensional vector made exclusively out of ones. In our numerical experiment we will always fall in the situation contemplated in (5.3) and it is this expression that we will use to construct the dynamic variance minimizing portfolios associated to each of the different models that we are testing. Figure 5.2 shows the conditional variance of the net returns process associated to the variance minimizing portfolios corresponding to the different models under consideration. It is tempting to say that the most performing model is the one for which the conditional variance is consistently smaller; however, given that the conditional variance is model dependent, *these quantities are not directly comparable* and it is only the marginal variances of the optimal portfolios that can be put side to side. This comparison is carried out in table 5.2 in which we see that VEC allows the construction of portfolios with smaller variances than those corresponding to the other models in all the dimensions considered.

	Variance of optimal portfolios ($\times 10^{-4}$)						
	$n=2$	$n=3$	$n=4$	$n=5$	$n=6$	$n=7$	$n=8$
EWMA	5.03	1.72	1.50	1.44	1.49	1.59	1.62
OGARCH	5.29	1.75	1.50	1.42	1.42	1.45	1.50
DCC	4.96	1.74	1.47	1.37	1.38	1.40	1.43
VEC	**4.91**	**1.70**	**1.39**	**1.22**	**1.15**	**1.15**	**1.12**

Table 5.2: Marginal variance of the net returns associated to the variance minimizing portfolios corresponding to the different models.

- **Goodness of fit between the associated conditional volatilities and the absolute values of the log-returns used as a proxy for conditional volatility:** following [MZ69, GN86, AB97, MCV02], we evaluate the performance of the different models by considering the absolute values of portfolio returns as a proxy for conditional volatility and by checking how the different proposals coming from the models under scrutiny fit this proxy. Even though it is well known [AB97] that this is a very noisy proxy for volatility, this approach provides us with quick and simple to implement ways to compare different modeling approaches. The first one consists of fitting each of the models to the first i assets with $i \in \{1, 2, \ldots, 8\}$ and computing the mean Euclidean distance (MSE) between the model associated conditional volatility and the proxy values; the results of this experiment are presented in table 5.3 where we see that VEC produces a smaller MSE in all the dimensions considered, even surprisingly at dimensions 7 and 8 where the number of parameters to be estimated is bigger than the sample size. The plausibility of this result is visually emphasized in figure 5.3 where we have depicted the conditional volatilities of one of the assets in our data set (AA) obtained out of the models under consideration in dimension 8, as well as of a one-dimensional GARCH model; these volatilities are graphically compared with the proxy.

Finally, we have ranked the different models by studying their efficiency in modeling the volatility of constant random portfolios; more especifically, at each dimension i, $i \in \{1, 2, \ldots, 8\}$, we randomly choose i weights using standard normally distributed variables and we appropriately normalize

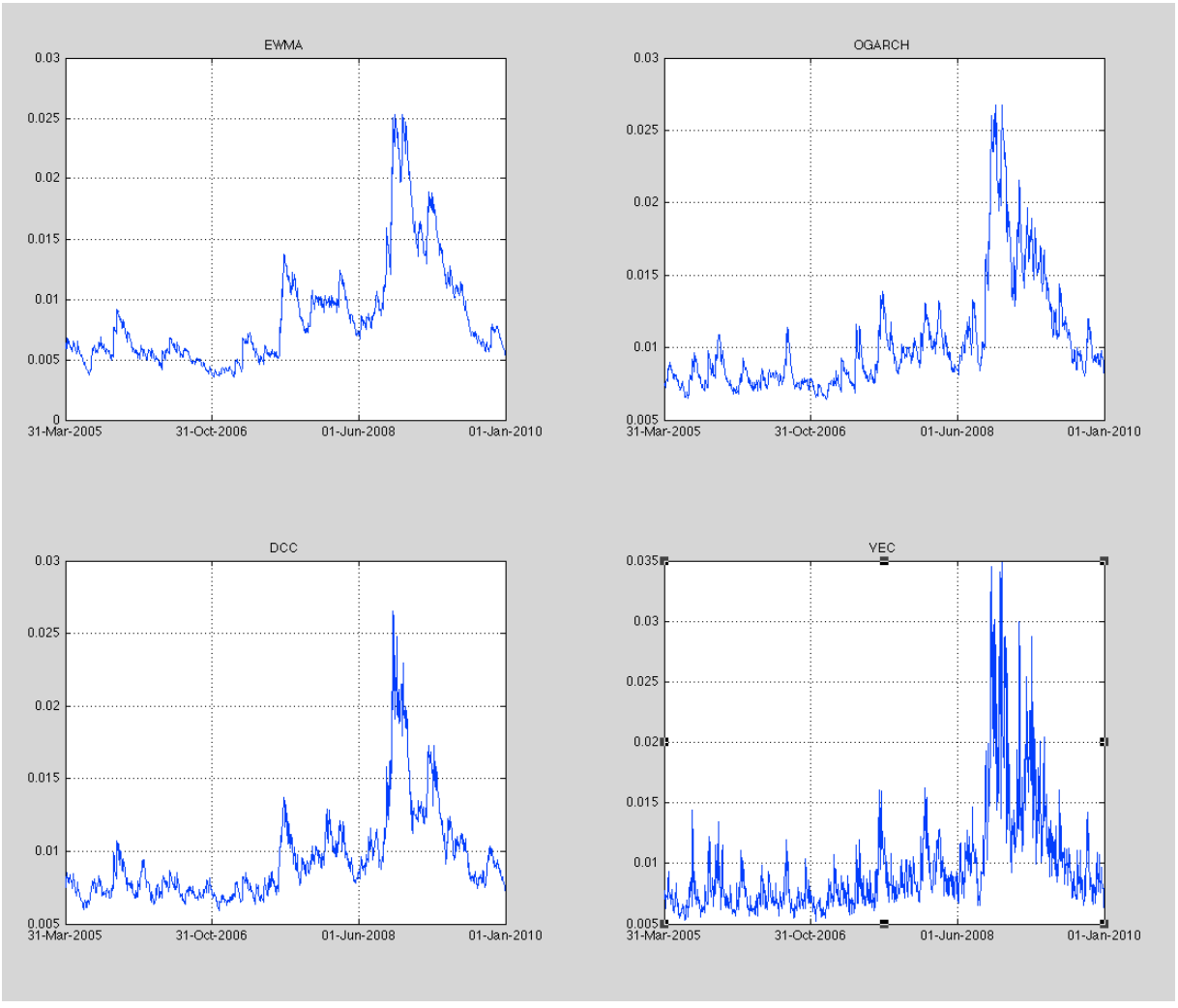

Figure 5.2: Conditional volatility of the net returns associated to the dynamic variance minimizing portfolios constructed by fitting different models to the log-returns of our eight dimensional data set. The VEC estimation was carried out setting a termination tolerance on the function value equal to 10^{-5} and using an OGARCH based OLS preliminary estimation, as explained in Section 4.4.

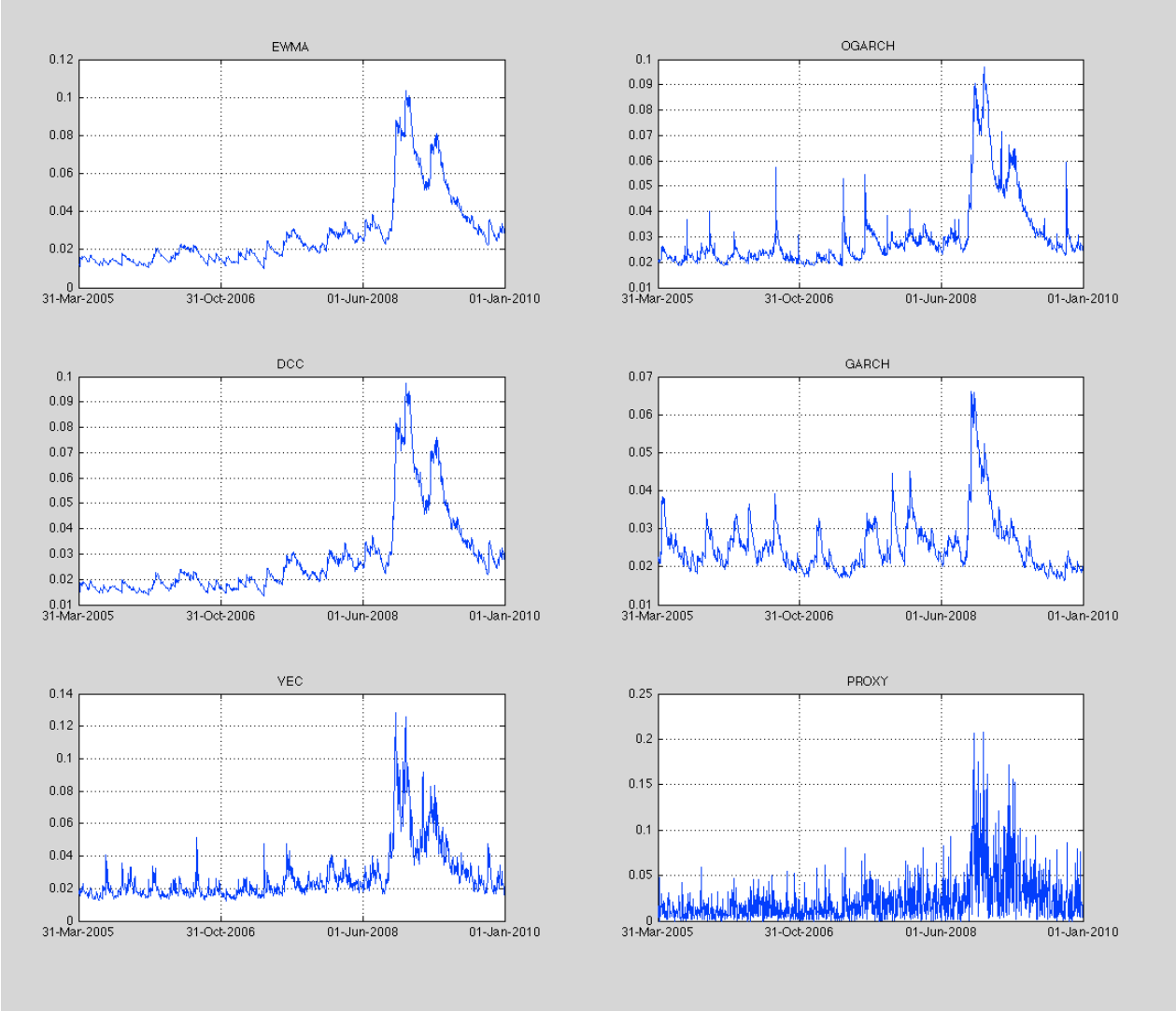

Figure 5.3: Conditional volatility of the asset Alcoa (AA) obtained out of eight dimensional modelings. The graphics EWMA, OGARCH, DCC, and VEC represent the volatility obtained as the square root of the $(1,1)$ components of the 8×8 conditional covariance matrices associated to those models. GARCH represents the volatility associated to a one-dimensional GARCH modeling of the log-returns of AA, and PROXY shows the absolute value of the AA log-returns.

	Mean square error with respect to proxy ($\times 10^{-4}$)							
	$n=1$	$n=2$	$n=3$	$n=4$	$n=5$	$n=6$	$n=7$	$n=8$
EWMA	5.19	4.31	3.23	2.71	3.03	2.89	3.42	3.39
OGARCH	**5.08**	4.33	3.24	2.71	3.02	2.88	3.44	3.38
DCC	**5.08**	4.23	3.17	2.66	2.98	2.85	3.39	3.43
VEC	**5.08**	**4.17**	**3.12**	**2.59**	**2.91**	**2.68**	**3.17**	**3.16**

Table 5.3: Mean square errors committed when modeling the absolute values of the returns with the conditional variance associated to the different models.

them so that their sum equals to one. We then use the conditional covariance matrices provided by each of the models under consideration to compute the (model based) conditional volatility of the portfolio. We then regress the proxy for the portfolio volatility, namely the absolute value of the portfolio returns, on the various portfolio volatilities provided by the different models and, using the suggestion in [MCV02] we declare as the best model the one that produces the highest coefficient of determination R^2. As the chosen proxy is know to be very noisy [AB97] the obtained R^2 coefficients are rather small (typically between 0.2 and 0.3). Using this criterion, we randomly generated 5,000 portfolios at each dimension, and we recorded the percentage rate of relative success of each model with respect to the others. The results of the experiment are presented in table 5.4 and show the superiority of VEC in all the dimensions considered.

	Success rate in modeling random portfolios (%)						
Number of assets	n=8	n=7	n=6	n=5	n=4	n=3	n=2
Number of VEC parameters	2628	1596	903	465	210	78	21
DCC	25.70	32.20	32.00	32.27	17.70	24.10	24.50
EWMA	1.27	0.9	0	0	0	0.3	0
OGARCH	28.18	19.40	8.30	7.87	3.70	21.80	22.80
VEC	**44.85**	**47.50**	**59.70**	**59.83**	**78.60**	**53.80**	**52.70**

Table 5.4: Percentage rate of relative success of each model with respect to the others in modeling the volatility of random portfolios. For each dimension n, we randomly generated 5,000 portfolios and we considered as the best model the one that produced the highest coefficient of determination R^2 when regressing the absolute values of the corresponding returns on the conditional covariances associated to the each model. VEC consitently presents the highest success rate regardless of the dimension.

- **Spectral sparsity and high dimensional estimation:** a major surprise revealed by these numerical experiments is that the estimated models provide good empirical performance despite the highly unfavorable ratio between the sample size and the number of parameters to be estimated. In order to investigate the reasons for such a counterintuitive but pleasing phenomenon, we plotted the eigenvalues of $\Sigma(A)$ and $\Sigma(B)$ for n between 4 to 8. These plots, displayed in Figure 5.4, show that the estimators $\Sigma(\hat{A})$ and $\Sigma(\hat{B})$ of $\Sigma(A)$ and $\Sigma(B)$ are spectrally very sparse, that is, have a very low rank. Thus, the solutions of the estimation problem are exactly the same as the ones we would have obtained under additional a priori rank constraints, a setting that would have implicitly reduced the dimension of the parameter space by a large factor. This suggests that in our particular empirical situation, the number of parameters that are actually independent is much smaller than the number of entries in the coefficient matrices A, B and $\mathbf{c}$, which makes

possible the use of small relative sample sizes in the VEC context. The most obvious explanation for this phenomenon stems from the well-known fact that the conditional covariance matrices H_t corresponding to stock market returns present spectral accumulation (in small dimensional settings) or sparsity (in large dimensions). This seems to make the positive semi-definiteness constraints on $\Sigma(A)$ and $\Sigma(B)$ highly active, which enforces a large number of eigenvalues to be equal to zero.

Notice further that the proportion of nonzero eigenvalues of $\Sigma(\hat{A})$ and $\Sigma(\hat{B})$ decreases very slowly as a function of the parameter space dimension and shows no particular abrupt transition when the dimension/sample size ratio becomes large. This phenomenon suggests that the constrained maximum likelihood approach is very stable for this hard estimation problem. On the other hand, pushing theses ideas further along the lines of recent works in sparse estimation and matrix completion problems [CR09, CT10], one might expect that explicitly enforcing the spectral sparsity of the estimators might improve their performance for dimensions much larger than the ones explored in the present work. A rigorous treatment of these observations is needed and will be the subject of further research in a forthcoming paper.

6 Conclusions

In this paper we provided an adequate explicit formulation of the estimation problem for VEC models and developed a Bregman-proximal trust-region method to solve it. This combination of techniques provides a robust optimization method that can be surely adapted with good results to more parsimonious multivariate volatility models.

We carried out numerical experiments based on stock market returns that show the applicability of the proposed estimation method in specific practical situations. Additionally, our numerical experiments reveal how the empirically well documented spectral accumulation in the covariance structure of stock quotes implies, in the context of VEC modeling, implicit nonlinear constraints in the parameter space that make this parametric family competitive even in the presence of a highly unfavorable ratio between the sample size and the number of parameters to be estimated. The comparison has been carried out with respect to other standard and more parsimonious multivariate conditionally heteroscedastic families, namely, EWMA, DCC, and OGARCH. An in-depth study of this phenomenon will be the subject of a forthcoming publication.

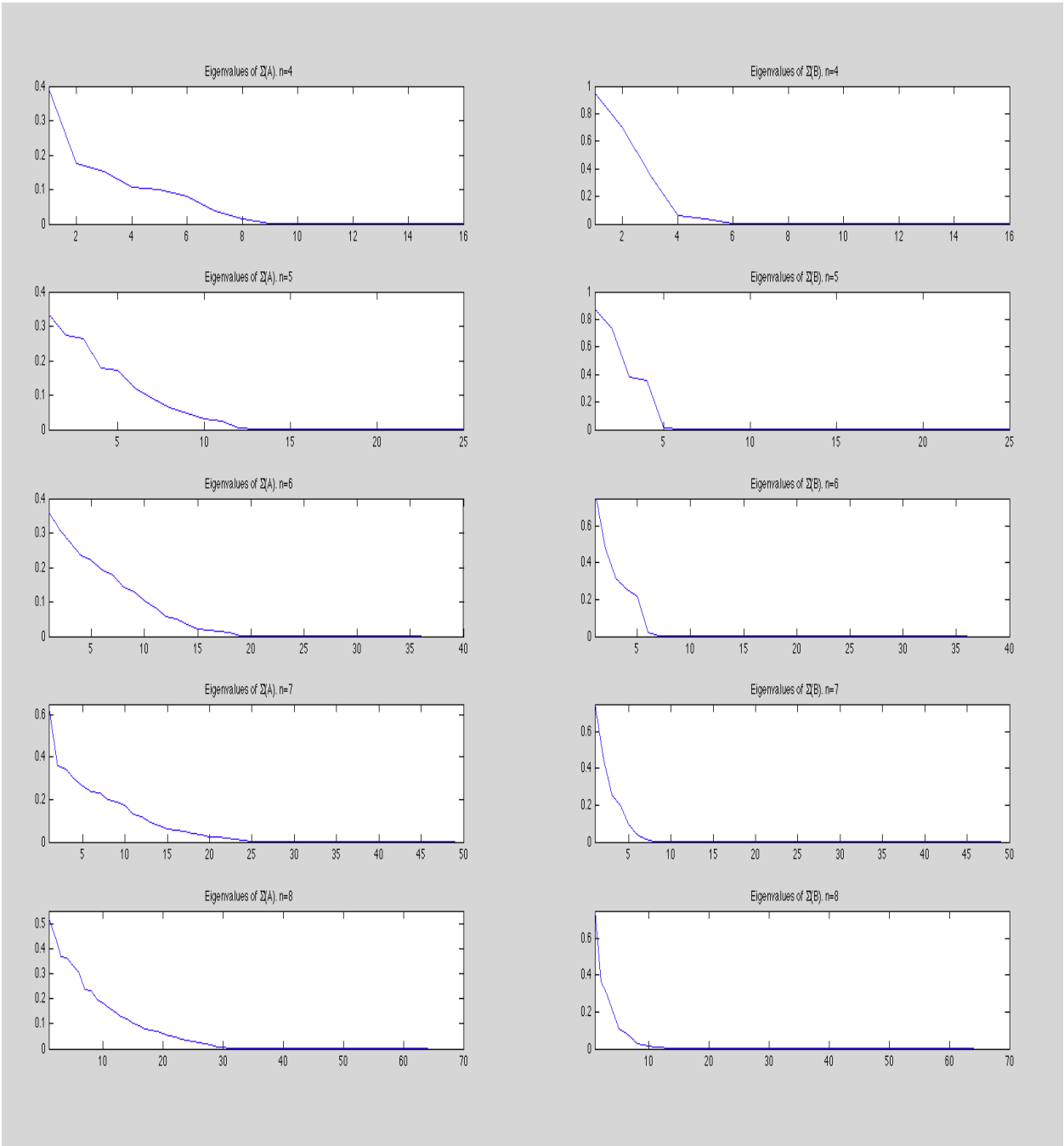

Figure 5.4: Eigenvalues of $\Sigma(A)$ and $\Sigma(B)$ for n between 4 to 8. The spectral sparsity evidenced in these plots suggests nonlinear constraints in the parameter space which explain the good empirical performance of the models despite the highly unfavorable ratio between the sample size and the number of parameters to be estimated.

7 Appendix

7.1 Proof of Proposition 2.1

We start with the proof of **(i)** by using the following chain of equalities in which we use the symmetric character of both A and $\mathrm{math}(m)$:

$$\begin{aligned}
\langle A + \mathrm{diag}(A), \mathrm{math}(m)\rangle &= \mathrm{trace}(A\,\mathrm{math}(m)) + \mathrm{trace}(\mathrm{diag}(A)\mathrm{math}(m)) \\
&= \sum_{i,j=1}^{n} A_{ij}\mathrm{math}(m)_{ji} + A_{ij}\delta_{ij}\mathrm{math}(m)_{ji} \\
&= \sum_{i<j} A_{ij}\mathrm{math}(m)_{ij} + \sum_{i>j} A_{ij}\mathrm{math}(m)_{ij} + 2\sum_{i=j=1}^{n} A_{ij}\mathrm{math}(m)_{ij} \\
&= 2\sum_{i\geq j} A_{ij}\mathrm{math}(m)_{ij} = 2\sum_{i\geq j} A_{ij}m_{\sigma(i,j)} = 2\sum_{q=1}^{N} A_{\sigma^{-1}(q)}m_q \\
&= 2\langle \mathrm{vech}(A), m\rangle,
\end{aligned}$$

as required. In order to prove **(ii)**, note that the identity that we just showed ensures that

$$\langle A, \mathrm{math}(m)\rangle = 2\langle \mathrm{vech}(A), m\rangle - \langle \mathrm{diag}(A), \mathrm{math}(m)\rangle. \tag{7.1}$$

At the same time

$$\begin{aligned}
\langle \mathrm{diag}(A), \mathrm{math}(m)\rangle &= \mathrm{trace}(\mathrm{diag}(A)\mathrm{math}(m)) = \sum_{i=1}^{n} A_{ii}\mathrm{math}(m)_{ii} = \sum_{i=1}^{n} A_{ii}m_{\sigma(i,i)} \\
&= \sum_{i\geq j} \mathrm{diag}(A)_{ij}m_{\sigma(i,j)} = \sum_{q=1}^{N} \mathrm{diag}(A)_{\sigma^{-1}(q)}m_q \\
&= \sum_{q=1}^{N} \mathrm{vech}(\mathrm{diag}(A))_q m_q = \langle \mathrm{vech}(\mathrm{diag}(A)), m\rangle,
\end{aligned}$$

which substituted in the right hand side of (7.1) proves the required identity. Finally, expression (2.3) follows directly from **(ii)** and as to (2.4) we observe that

$$\begin{aligned}
\frac{1}{2}\langle A + \mathrm{diag}(A), \mathrm{math}(m)\rangle &= \frac{1}{2}\,\mathrm{trace}((A + \mathrm{diag}(A))\mathrm{math}(m)) \\
&= \frac{1}{2}\,\mathrm{trace}(A\mathrm{math}(m)) + \frac{1}{2}\,\mathrm{trace}(\mathrm{diag}(A)\mathrm{math}(m)) \\
&= \frac{1}{2}\,(\mathrm{trace}(A\mathrm{math}(m)) + \mathrm{trace}(A\mathrm{diag}(\mathrm{math}(m)))) \\
&= \frac{1}{2}\langle A, \mathrm{math}(m) + \mathrm{diag}(\mathrm{math}(m))\rangle,
\end{aligned}$$

which proves (2.4). Regarding the operator norms we will just prove (2.5) and (2.6) as the rest can be easily obtained out of these two combined with the expressions (2.3) and (2.4). We start by noticing that for any nonzero $A = (a_{ij}) \in \mathbb{S}_n$:

$$\frac{\|\mathrm{vech}(A)\|^2}{\|A\|^2} = \frac{\sum_{i>j=1}^{n} a_{ij}^2 + \sum_{i=1}^{n} a_{ii}^2}{2\sum_{i>j=1}^{n} a_{ij}^2 + \sum_{i=1}^{n} a_{ii}^2} = 1 - \frac{\sum_{i>j=1}^{n} a_{ij}^2}{2\sum_{i>j=1}^{n} a_{ij}^2 + \sum_{i=1}^{n} a_{ii}^2}.$$

Since the last summand in the previous expression is always positive we have that

$$\|\text{vech}\|_{op} = \sup_{A\in\mathbb{S}_n, A\neq 0} \frac{\|\text{vech}(A)\|}{\|A\|} = 1,$$

the supremum being attained by any diagonal matrix ($\sum_{i>j=1}^n a_{ij}^2 = 0$ in that case). Consider now $v = \text{vech}(A)$. Then:

$$\frac{\|\text{math}(v)\|^2}{\|v\|^2} = \frac{\|A\|^2}{\|\text{vech}(A)\|^2} = \frac{2\sum_{i>j=1}^n a_{ij}^2 + \sum_{i=1}^n a_{ii}^2}{\sum_{i>j=1}^n a_{ij}^2 + \sum_{i=1}^n a_{ii}^2} = 1 + \frac{\sum_{i>j=1}^n a_{ij}^2}{\sum_{i>j=1}^n a_{ij}^2 + \sum_{i=1}^n a_{ii}^2}. \tag{7.2}$$

When we let $A \in \mathbb{S}_n$ vary in the previous expression, we obtain a supremum by considering matrices with zeros in the diagonal ($\sum_{i=1}^n a_{ii}^2 = 0$) and by choosing $\sum_{i>j=1}^n a_{ij}^2 \to \infty$, in which case $\frac{\|A\|^2}{\|\text{vech}(A)\|^2} \to 2$. Finally, as the map $\text{vech} : \mathbb{S}_n \to \mathbb{R}^N$ is an isomorphism, (7.2) implies that

$$\|\text{math}\|_{op} = \sup_{v\in\mathbb{R}^N, v\neq 0} \frac{\|\text{math}(v)\|}{\|v\|} = \sup_{A\in\mathbb{S}_n, A\neq 0} \frac{\|A\|}{\|\text{vech}(A)\|} = \sqrt{2}. \quad \blacksquare$$

7.2 Proof of Proposition 2.3

We just need to verify that (2.11) satisfies (2.12). Let $k, l \in \{1, \ldots, n\}$ be such that $k \geq l$. Then,

$$\begin{aligned}
(A\,\text{vech}(H))_{\sigma(k,l)} &= \sum_{i\geq j} A_{\sigma(k,l),\sigma(i,j)} H_{ij} = \sum_{i\geq j} A_{\sigma(k,l),\sigma(i,j)} \frac{H_{ij} + H_{ji}}{2} \\
&= \frac{1}{2}\sum_{i\geq j} A_{\sigma(k,l),\sigma(i,j)} H_{ij} + \frac{1}{2}\sum_{i\geq j} A_{\sigma(k,l),\sigma(i,j)} H_{ji} \\
&= \frac{1}{2}\sum_{i>j} A_{\sigma(k,l),\sigma(i,j)} H_{ij} + \sum_{i=j} A_{\sigma(k,l),\sigma(i,j)} H_{ij} + \frac{1}{2}\sum_{i<j} A_{\sigma(k,l),\sigma(j,i)} H_{ij} \\
&= \sum_{i>j} (\Sigma(A)_{kl})_{ij} H_{ij} + \sum_{i=j} (\Sigma(A)_{kl})_{ij} H_{ij} + \sum_{i<j} (\Sigma(A)_{kl})_{ij} H_{ij} = \text{trace}(\Sigma(A)_{kl} H),
\end{aligned}$$

as required. $\blacksquare$

7.3 Proof of Proposition 2.4

We start with the following Lemma:

Lemma 7.1 *Let $A \in \mathbb{M}_{n^2}$. The orthogonal projections $\mathbb{P}_{n^2}(A) \in \mathbb{S}_{n^2}$ and $\mathbb{P}^n_{n^2}(A) \in \mathbb{S}^n_{n^2}$ of A onto the spaces of symmetric and n-symmetric matrices with respect to the Frobenius inner product (2.1) are given by:*

$$\mathbb{P}_{n^2}(A) = \frac{1}{2}(A + A^T) \tag{7.3}$$

$$(\mathbb{P}^n_{n^2}(A))_{kl} = \frac{1}{4}(A_{kl} + A_{kl}^T + A_{lk} + A_{lk}^T), \tag{7.4}$$

for any block $(\mathbb{P}^n_{n^2}(A))_{kl}$ of $\mathbb{P}^n_{n^2}(A)$, $k, l \in \{1, \ldots, n\}$.

Proof. In order to prove (7.3) it suffices to check that $\langle A - \mathbb{P}_{n^2}(A), B\rangle = 0$ for any $B \in \mathbb{S}_{n^2}$. Indeed,

$$\langle A - \mathbb{P}_{n^2}(A), B\rangle = \operatorname{trace}(AB) - \frac{1}{2}\operatorname{trace}(AB) - \frac{1}{2}\operatorname{trace}(A^T B) = 0.$$

The result follows from the uniqueness of the orthogonal projection. Regarding (7.4) we check that $\langle A - \mathbb{P}^n_{n^2}(A), B\rangle = 0$, for any $B \in \mathbb{S}^n_{n^2}$. Given that for any $k, l \in \{1, \dots, n\}$ the block $(AB)_{kl}$ is given by $(AB)_{kl} = \sum_{r=1}^n A_{kr}B_{rl}$ we have

$$\begin{aligned}
\langle A - \mathbb{P}^n_{n^2}(A), B\rangle &= \operatorname{trace}(AB) - \operatorname{trace}(\mathbb{P}^n_{n^2}(A)B) = \sum_{i=1}^n \operatorname{trace}(AB)_{ii} - \operatorname{trace}(\mathbb{P}^n_{n^2}(A)B)_{ii} \\
&= \sum_{i,j=1}^n \operatorname{trace}(A_{ij}B_{ji}) - \operatorname{trace}((\mathbb{P}^n_{n^2}(A))_{ij}B_{ji}) = \sum_{i,j=1}^n \operatorname{trace}(A_{ij}B_{ji}) \\
&- \sum_{i,j=1}^n \left[\frac{1}{4}\operatorname{trace}(A_{ij}B_{ji}) + \frac{1}{4}\operatorname{trace}(A^T_{ij}B_{ji}) + \frac{1}{4}\operatorname{trace}(A_{ji}B_{ji}) + \frac{1}{4}\operatorname{trace}(A^T_{ji}B_{ji})\right] = 0,
\end{aligned}$$

where we used that, due to the n-symmetricity of B $\operatorname{trace}(A^T_{ij}B_{ji}) = \operatorname{trace}(B^T_{ji}A_{ij}) = \operatorname{trace}(A_{ij}B_{ji})$ and

$$\sum_{i,j=1}^n \operatorname{trace}(A_{ji}B_{ji}) = \operatorname{trace}(A_{ji}B_{ij}) = \operatorname{trace}(A_{ij}B_{ij}).$$

Analogously $\sum_{i,j=1}^n \operatorname{trace}(A^T_{ji}B_{ji}) = \operatorname{trace}(A_{ij}B_{ij})$. ■

Now, in order to prove Proposition 2.4, consider $A \in \mathbb{M}_N$ and $\mathcal{B} \in \mathbb{M}_{n^2}$. Since the image of the map Σ lies in $\mathbb{S}^2_{n^2}$ we have that $\langle \mathcal{B} - \mathbb{P}^n_{n^2}(\mathcal{B}), \Sigma(A)\rangle = 0$ and hence

$$\langle \Sigma^*(\mathcal{B}), A\rangle = \langle \mathcal{B}, \Sigma(A)\rangle = \langle \mathbb{P}^n_{n^2}(\mathcal{B}) + \mathcal{B} - \mathbb{P}^n_{n^2}(\mathcal{B}), \Sigma(A)\rangle = \langle \mathbb{P}^n_{n^2}(\mathcal{B}), \Sigma(A)\rangle = \langle \Sigma^*(\mathbb{P}^n_{n^2}(\mathcal{B})), A\rangle.$$

This identity allows us to restrict the proof of (2.4) to the n-symmetric elements $\mathcal{B} \in \mathbb{S}^n_{n^2}$. Hence let $\mathcal{B} \in \mathbb{S}^n_{n^2}$ and let $\widetilde{\sigma}$ be the extension of the map σ defined in (2.2). Then,

$$\begin{aligned}
\langle \Sigma(A), \mathcal{B}\rangle &= \sum_{k,l=1}^n \langle \Sigma(A)_{kl}, \mathcal{B}_{kl}\rangle = \sum_{k,l=1}^n \operatorname{trace}(\Sigma(A)_{kl}\mathcal{B}^T_{kl}) = \sum_{k,l,i,j=1}^n (\Sigma(A)_{kl})_{ij}(\mathcal{B}_{kl})_{ij} \\
&= \sum_{k,l,i,j=1}^n \frac{1}{2}\left[A_{\widetilde{\sigma}(k,l),\widetilde{\sigma}(i,j)} + A_{\widetilde{\sigma}(k,l),\widetilde{\sigma}(i,j)}\delta_{ij}\right](\mathcal{B}_{kl})_{ij} \\
&= \sum_{k,j=1}^n \left[\sum_{i<j}^n \frac{1}{2}A_{\widetilde{\sigma}(k,l),\sigma(j,i)}(\mathcal{B}_{kl})_{ji} + \sum_{i=j=1}^n A_{\widetilde{\sigma}(k,l),\sigma(i,j)}(\mathcal{B}_{kl})_{ij} + \frac{1}{2}\sum_{i>j}^n A_{\widetilde{\sigma}(k,l),\sigma(i,j)}(\mathcal{B}_{kl})_{ij}\right] \\
&= \sum_{k,j=1}^n \sum_{i\geq j}^n A_{\widetilde{\sigma}(k,l),\sigma(i,j)}(\mathcal{B}_{kl})_{ji} \\
&= \sum_{i\geq j}^n \left[\sum_{k<l}^n A_{\sigma(l,k),\sigma(j,i)}(\mathcal{B}_{lk})_{ji} + \sum_{k=l=1}^n A_{\sigma(k,l),\sigma(i,j)}(\mathcal{B}_{kl})_{ij} + \sum_{l<k}^n A_{\sigma(k,l),\sigma(i,j)}(\mathcal{B}_{kl})_{ij}\right] \\
&= \sum_{i\geq j}^n \left[\sum_{k\geq l}^n A_{\sigma(k,l),\sigma(i,j)}(\mathcal{B}_{kl})_{ij} - \sum_{k=l=1}^n A_{\sigma(k,l),\sigma(i,j)}(\mathcal{B}_{kl})_{ij}\delta_{kl}\right] \\
&= \sum_{p,q=1}^N \left[2A_{p,q}B_{p,q} - A_{p,q}B_{p,q}\delta_{\mathrm{pr}_1(\sigma^{-1}(p)),\mathrm{pr}_2(\sigma^{-1}(p))}\right] = \operatorname{trace}(2AB^T - A\widetilde{B}^T) = \langle A, 2B - \widetilde{B}\rangle,
\end{aligned}$$

which proves the statement. We emphasize that in the fourth and sixth equalities we used the n-symmetry of $\mathcal{B}$. The equality (2.14) is proved in a straightforward manner by verifying that $\widetilde{\Sigma}^{-1} \circ \Sigma = \mathbb{I}_{\mathbb{M}_N}$ and $\Sigma \circ \widetilde{\Sigma}^{-1} = \mathbb{I}_{\mathbb{S}^n_{n^2}}$ using the defining expressions (2.2) and (2.14). ■

7.4 Proof of Proposition 3.1

Using the property of the operator Σ stated in Proposition 2.3, the second equality in (3.1) can be rewritten as:

$$\begin{aligned} \operatorname{vech}(H_t) &= \operatorname{vech}(\operatorname{math}(\mathbf{c})) + A\operatorname{vech}(\mathbf{z}_{t-1}\mathbf{z}_{t-1}^T) + B\operatorname{vech}(H_{t-1}) \\ &= \operatorname{vech}(\operatorname{math}(\mathbf{c})) + \operatorname{vech}(\Sigma(A) \bullet (\mathbf{z}_{t-1}\mathbf{z}_{t-1}^T)) + \operatorname{vech}(\Sigma(B) \bullet H_{t-1}), \end{aligned}$$

or, equivalently:

$$H_t = \operatorname{math}(\mathbf{c}) + \Sigma(A) \bullet (\mathbf{z}_{t-1}\mathbf{z}_{t-1}^T) + \Sigma(B) \bullet H_{t-1}.$$

In view of this expression and in the terms of the statement of the proposition, it suffices to show that both $\Sigma(A) \bullet (\mathbf{z}_{t-1}\mathbf{z}_{t-1}^T)$ and $\Sigma(B) \bullet H_{t-1}$ are positive semidefinite provided that H_{t-1} is positive semidefinite. Regarding $\Sigma(A) \bullet (\mathbf{z}_{t-1}\mathbf{z}_{t-1}^T)$, consider $\mathbf{v} \in \mathbb{R}^{n^2}$. Then

$$\begin{aligned} \langle \mathbf{v}, \Sigma(A) \bullet (\mathbf{z}_{t-1}\mathbf{z}_{t-1}^T)\mathbf{v} \rangle &= \sum_{i,j=1}^{n^2} v_i (\Sigma(A) \bullet (\mathbf{z}_{t-1}\mathbf{z}_{t-1}^T))_{ij} v_j = \sum_{i,j=1}^{n^2} v_i \operatorname{trace}(\Sigma(A)_{ij}(\mathbf{z}_{t-1}\mathbf{z}_{t-1}^T)) v_j \\ &= \sum_{i,j=1}^{n^2} v_i \operatorname{trace}(\mathbf{z}_{t-1}^T \Sigma(A)_{ij} \mathbf{z}_{t-1}) v_j = \sum_{i,j,k,l=1}^{n^2} v_i z_{t-1,k}^T (\Sigma(A)_{ij})_{kl} z_{t-1,l} v_j \\ &= \langle \mathbf{v} \otimes \mathbf{z}_{t-1}, \Sigma(A)(\mathbf{v} \otimes \mathbf{z}_{t-1}) \rangle, \end{aligned}$$

which is greater or equal to zero due to the positive semidefiniteness hypothesis on $\Sigma(A)$. In the last equality we used (2.10).

As to $\Sigma(B) \bullet H_{t-1}$, we start by noticing that $H_{t-1} = E_{t-1}[\mathbf{z}_{t-1}\mathbf{z}_{t-1}^T]$ and hence $\Sigma(B) \bullet H_{t-1} = \Sigma(B) \bullet E_{t-1}[\mathbf{z}_{t-1}\mathbf{z}_{t-1}^T]$. This equality, as well as the linearity of the conditional expectation allows us to use virtually the same argument as above. Indeed, for any $\mathbf{v} \in \mathbb{R}^{n^2}$

$$\begin{aligned} \langle \mathbf{v}, \Sigma(B) \bullet H_{t-1}\mathbf{v} \rangle &= \sum_{i,j=1}^{n^2} v_i \operatorname{trace}(\Sigma(B)_{ij} E_{t-1}[\mathbf{z}_{t-1}\mathbf{z}_{t-1}^T]) v_j = \sum_{i,j=1}^{n^2} E_{t-1}[v_i \operatorname{trace}(\Sigma(B)_{ij}\mathbf{z}_{t-1}\mathbf{z}_{t-1}^T) v_j] \\ &= E_{t-1}[\langle \mathbf{v} \otimes \mathbf{z}_{t-1}, \Sigma(B)(\mathbf{v} \otimes \mathbf{z}_{t-1}) \rangle], \end{aligned}$$

which is greater or equal to zero due to the positive semidefiniteness hypothesis on $\Sigma(B)$. ■

7.5 Proof of Proposition 3.2

We start by noticing that the VEC(1,1) model is by construction a white noise and hence it suffices to establish the stationarity of the variance. Indeed, for any $t, h \in \mathbb{N}$ we compute the autocovariance function Γ:

$$\begin{aligned} \Gamma(t, t+h) := E\left[\mathbf{z}_t \mathbf{z}_{t+h}^T\right] &= E\left[E_t\left[H_t^{1/2} \boldsymbol{\epsilon_t} \boldsymbol{\epsilon_{t+h}} H_{t+h}^{1/2}\right]\right] \\ &= E\left[H_t^{1/2} E_t\left[\boldsymbol{\epsilon_t} \boldsymbol{\epsilon_{t+h}}\right] H_{t+h}^{1/2}\right] = \delta_{h0} E\left[H_t^{1/2} H_{t+h}^{1/2}\right]. \quad (7.5) \end{aligned}$$

Consequently, we just need to prove the existence of a solution for which $\Gamma(t,t) = E\left[H_t\right]$ or, equivalently $E[\mathbf{h}_t]$, is time independent. We first notice that

$$E[\mathbf{h}_t] = E\left[\mathbf{c} + A\boldsymbol{\eta}_{t-1} + B\boldsymbol{h}_{t-1}\right] = E\left[\mathbf{c} + A\boldsymbol{h}_{t-1} + B\boldsymbol{h}_{t-1}\right] + A\,E\left[\boldsymbol{\eta}_{t-1} - \boldsymbol{h}_{t-1}\right] = E\left[\mathbf{c} + A\boldsymbol{h}_{t-1} + B\boldsymbol{h}_{t-1}\right],$$

since $A\,E\left[\boldsymbol{\eta}_{t-1} - \boldsymbol{h}_{t-1}\right] = 0$ by (7.5). Now, for any $k > 0$

$$E[\mathbf{h}_t] = \mathbf{c} + (A+B)E\left[\mathbf{h}_{t-1}\right] = \sum_{j=0}^{k}(A+B)^j\mathbf{c} + (A+B)^{k+1}E\left[\mathbf{h}_{t-k-1}\right].$$

If all the eigenvalues of $A+B$ are smaller than one in modulus then (see, for example [Lüt05, Appendix A.9.1])

$$\sum_{j=0}^{k}(A+B)^j\mathbf{c} \xrightarrow[k\to\infty]{} (\mathbb{I}_N - A - B)^{-1}\mathbf{c}, \qquad \text{and} \qquad (A+B)^{k+1}E\left[\mathbf{h}_{t-k-1}\right] \xrightarrow[k\to\infty]{} 0,$$

in which case $E[\mathbf{h}_t]$ is time independent and

$$\Gamma(0) = \mathrm{math}(E[\mathbf{h}_t]) = \mathrm{math}((\mathbb{I}_N - A - B)^{-1}\mathbf{c}).$$

The sufficient condition in terms of the top singular value $\sigma_{\max}(A+B)$ of $A+B$ is a consequence of the fact that (see for instance [HJ94, Theorem 5.6.9]) $|\lambda(A+B)| \leq \sigma_{\max}(A+B)$, for any eigenvalue $\lambda(A+B)$ of $A+B$. ■

7.6 Proof of Proposition 3.3

The chain rule implies that for any perturbation Δ in the θ direction

$$d_\theta l_t \cdot \Delta = d_{H_t} l_t(H_t(\theta)) \cdot T_\theta H_t \cdot \Delta = \langle \nabla_{H_t} l_t, T_\theta H_t \cdot \Delta\rangle = \langle T_\theta^* H_t \cdot \nabla_{H_t} l_t, \Delta\rangle,$$

which proves that $\nabla_\theta l_t = T_\theta^* H_t \cdot \nabla_{H_t} l_t$ and hence (3.8) follows. We now establish (3.9) by showing separately that

$$\nabla_{H_t}\log(\det(H_t)) = H_t^{-1} \quad \text{and} \quad \nabla_{H_t}\left(-\frac{1}{2}\mathbf{z}_t^T H_t^{-1}\mathbf{z}_t\right) = \frac{1}{2}\left(H_t^{-1}\mathbf{z}_t\mathbf{z}_t^T H_t^{-1}\right). \tag{7.6}$$

In order to prove the first expression we start by using the positive semidefinite character of H_t in order to write $H_t = VDV^T$. V is an orthogonal matrix and D is diagonal with non-negative entries; it has hence a unique square root $D^{1/2}$ that we can use to write $H_t = VDV^T = (VD^{1/2})(VD^{1/2})^T$. Let $\delta \in \mathbb{R}$ and $\Delta \in \mathbb{S}_n$. We have

$$\begin{aligned}
\log(\det(H_t + \delta\Delta)) &= \log(\det((VD^{1/2})(VD^{1/2})^T + \delta\Delta)) \\
&= \log(\det((VD^{1/2})(\mathbb{I}_n + \delta(D^{-1/2}V^T)\Delta(VD^{-1/2}))(VD^{1/2})^T)) \\
&= \log(\det(VD^{1/2})\det(\mathbb{I}_n + \delta(D^{-1/2}V^T)\Delta(VD^{-1/2}))\det(VD^{1/2})^T) \\
&= \log(\det((VD^{1/2})(VD^{1/2})^T)\det(\mathbb{I}_n + \delta(D^{-1/2}V^T)\Delta(VD^{-1/2}))) \\
&= \log(\det(H_t)\det(\mathbb{I}_n + \delta\Xi)),
\end{aligned}$$

with $\Xi := (D^{-1/2}V^T)\Delta(VD^{-1/2})$. This matrix is symmetric and hence normal and diagonalizable; let $\{\lambda_1, \ldots, \lambda_n\}$ be its eigenvalues. We hence have that

$$\begin{aligned}
dH_t \cdot \Delta &= \left.\frac{d}{d\delta}\right|_{\delta=0}\log(\det(H_t + \delta\Delta)) = \left.\frac{d}{d\delta}\right|_{\delta=0}\log(\det(H_t)) + \log\left(\prod_{i=1}^{n}(1+\delta\lambda_i)\right) = \left.\frac{d}{d\delta}\right|_{\delta=0}\sum_{i=1}^{n}\log(1+\delta\lambda_i) \\
&= \sum_{i=1}^{n}\lambda_i = \mathrm{trace}((D^{-1/2}V^T)\Delta(VD^{-1/2})) = \mathrm{trace}((VD^{-1/2})(D^{-1/2}V^T)\Delta) = \mathrm{trace}(H_t^{-1}\Delta),
\end{aligned}$$

which proves $\nabla_{H_t} \log(\det(H_t)) = H_t^{-1}$. Regarding the second expression in (7.6) we define $f(H_t) := -\frac{1}{2}\mathbf{z}_t^T H_t^{-1}\mathbf{z}_t$ and note that

$$\begin{aligned} df(H_t)\cdot\Delta &= \frac{d}{dt}\Big|_{t=0} -\frac{1}{2}\mathbf{z}_t^T(H_t+t\Delta)^{-1}\mathbf{z}_t = \frac{d}{dt}\Big|_{t=0} -\frac{1}{2}\mathbf{z}_t^T(\mathbb{I}_n + tH_t^{-1}\Delta)^{-1}H_t^{-1}\mathbf{z}_t \\ &= \frac{d}{dt}\Big|_{t=0} -\frac{1}{2}\mathbf{z}_t^T(\mathbb{I}_n + tH_t^{-1}\Delta)^{-1}H_t^{-1}\mathbf{z}_t = \frac{d}{dt}\Big|_{t=0} -\frac{1}{2}\sum_{k=0}^{\infty}(-1)^k t^k \mathbf{z}_t^T(H_t^{-1}\Delta)^k H_t^{-1}\mathbf{z}_t \\ &= \frac{1}{2}\mathbf{z}_t^T H_t^{-1}\Delta H_t^{-1}\mathbf{z}_t = \frac{1}{2}\operatorname{trace}(H_t^{-1}\mathbf{z}_t\mathbf{z}_t^T H_t^{-1}\Delta), \end{aligned}$$

which implies that $\nabla_{H_t} f = \frac{1}{2}\left(H_t^{-1}\mathbf{z}_t\mathbf{z}_t^T H_t^{-1}\right)$, as required.

In order to prove (3.10)–(3.12) we notice that the second equation in (3.1) can be rewritten using the vech and math operators as

$$H_t = \operatorname{math}\left(\mathbf{c} + A\boldsymbol{\eta}_{t-1} + B\operatorname{vech}(H_{t-1})\right). \tag{7.7}$$

We now show (3.10). Let $\mathbf{v}\in\mathbb{R}^N$ and $\Delta\in\mathbb{S}_n$ arbitrary. Identity (7.7) and the linearity of the various mappings involved imply that $T_{\mathbf{c}}H_t\cdot\mathbf{v} = \operatorname{math}\left(\mathbf{v} + B\operatorname{vech}(T_{\mathbf{c}}H_{t-1}\cdot\mathbf{v})\right)$ and hence

$$\begin{aligned} \langle T_{\mathbf{c}}^* H_t\cdot\Delta, \mathbf{v}\rangle &= \langle\Delta, T_{\mathbf{c}}H_t\cdot\mathbf{v}\rangle = \langle\Delta, \operatorname{math}\left(\mathbf{v} + B\operatorname{vech}(T_{\mathbf{c}}H_{t-1}\cdot\mathbf{v})\right)\rangle \\ &= \langle \operatorname{math}^*(\Delta) + T_{\mathbf{c}}^* H_{t-1}\cdot\operatorname{vech}^*(B^T\operatorname{math}^*(\Delta)), \mathbf{v}\rangle. \end{aligned}$$

The proof of (3.11) follows a similar scheme. By (7.7) we have that for any $M\in\mathbb{M}_N$:

$$T_A H_t\cdot M = \operatorname{math}\left(M\boldsymbol{\eta}_{t-1} + B\operatorname{vech}(T_A H_{t-1}\cdot M)\right). \tag{7.8}$$

Consequently, for any $\Delta\in\mathbb{S}_n$

$$\begin{aligned} \langle T_A^* H_t\cdot\Delta, M\rangle &= \langle\Delta, T_A H_t\rangle = \langle\Delta, \operatorname{math}\left(M\boldsymbol{\eta}_{t-1} + B\operatorname{vech}(T_A H_{t-1}\cdot M)\right)\rangle \\ &= \langle \operatorname{math}^*(\Delta)\cdot\boldsymbol{\eta}_{t-1}^T + T_A^* H_{t-1}\cdot\operatorname{vech}^*(B^T\operatorname{math}^*(\Delta)), \Delta\rangle. \end{aligned}$$

Finally, (3.12) is proved analogously replacing (7.8) by its B counterpart, namely,

$$T_B H_t\cdot M = \operatorname{math}\left(M\operatorname{vech}(H_{t-1}) + B\operatorname{vech}(T_B H_{t-1}\cdot M)\right). \quad \blacksquare$$

7.7 Proof of Proposition 3.4

An inductive argument using (3.10)–(3.12) guarantees that for any $t,k\in\mathbb{N}$, $k\le t$

$$T_{\mathbf{c}}H_t^*\cdot\Delta = \sum_{i=1}^{k} B^{i-1\,T}\operatorname{math}^*(\Delta) + T_{\mathbf{c}}^* H_{t-k}\cdot\operatorname{vech}^*(B^{k\,T}\operatorname{math}^*(\Delta)), \tag{7.9}$$

$$T_A^* H_t\cdot\Delta = \sum_{i=1}^{k} B^{i-1\,T}\operatorname{math}^*(\Delta)\cdot\boldsymbol{\eta}_{t-i}^T + T_A^* H_{t-k}\cdot\operatorname{vech}^*(B^{k\,T}\operatorname{math}^*(\Delta)), \tag{7.10}$$

$$T_B^* H_t\cdot\Delta = \sum_{i=1}^{k} B^{i-1\,T}\operatorname{math}^*(\Delta)\cdot\operatorname{vech}(H_{t-i})^T + T_B^* H_{t-k}\cdot\operatorname{vech}^*(B^{k\,T}\operatorname{math}^*(\Delta)), \tag{7.11}$$

The first expression with $k=t$ and the norm estimate (2.8) imply that

$$\|T_{\mathbf{c}}H_t^*\cdot\Delta\| = \left\|\sum_{i=1}^{t} B^{i-1\,T}\operatorname{math}^*(\Delta)\right\| \le \sqrt{2}\sum_{i=1}^{t}\|B\|_{\mathrm{op}}^{i-1}\|\Delta\| \le \frac{\sqrt{2}\|\Delta\|}{1-\|B\|_{\mathrm{op}}}. \tag{7.12}$$

We now use (7.9) for an arbitrary k as well as (2.7) and (7.12) and write

$$\|(T^*_{\mathbf{c}}H_t - T^*_{\mathbf{c}}H^k_t)\cdot\Delta\| = \|T^*_{\mathbf{c}}H_{t-k}\cdot \mathrm{vech}^*(B^{k\,T}\mathrm{math}^*(\Delta))\|$$
$$\leq \|T^*_{\mathbf{c}}H_{t-k}\|_{\mathrm{op}}\|\mathrm{vech}^*\|_{\mathrm{op}}\|B\|^k_{\mathrm{op}}\|\mathrm{math}^*\|_{\mathrm{op}}\|\Delta\| \leq \frac{2\|\Delta\|\|B\|^k_{\mathrm{op}}}{1-\|B\|_{\mathrm{op}}}. \quad (7.13)$$

The computability constraint **(CC)** implies that $\|B\|_{\mathrm{op}} \leq 1-\widetilde{\epsilon}_B$ and hence $\|T^*_{\mathbf{c}}H_t - T^*_{\mathbf{c}}H^k_t\|_{\mathrm{op}} \leq 2(1-\widetilde{\epsilon}_B)^k/\widetilde{\epsilon}_B$. A straightforward computation shows that if we want this upper bound for the error to be smaller than a certain $\delta>0$, that is $2(1-\widetilde{\epsilon}_B)^k/\widetilde{\epsilon}_B<\delta$ then it suffices to take

$$k > \frac{\log\left(\frac{\widetilde{\epsilon}_B\delta}{2}\right)}{\log(1-\widetilde{\epsilon}_B)}. \quad (7.14)$$

We now tackle the estimation of the truncation error in mean in the A variable. Firstly, we recall that by (7.5) and in the presence of the stationarity constraint $E[\boldsymbol{\eta}_t] = E[\mathbf{h}_t] = (\mathbb{I}_N - A - B)^{-1}\mathbf{c}$. The first consequence of this identity is that if we take the expectations of both (7.10) and (7.11) we see that $\|E\left[T^*_A H_t\cdot\Delta\right]\|$ and $\|E\left[T^*_B H_t\cdot\Delta\right]\|$ are determined by exactly the same recursions and hence the error estimations for both variables are going to be the same. Also, by (7.10)

$$\|E\left[T^*_A H_t\cdot\Delta\right]\| = \left\|\sum_{i=1}^{t} B^{i-1\,T}\mathrm{math}^*(\Delta)\cdot E[\boldsymbol{\eta}^T_{t-i}]\right\| \leq \sqrt{2}\|\Delta\|\|E[\mathbf{h}_t]\|\sum_{i=1}^{t}\|B\|^{i-1}_{\mathrm{op}}$$
$$\leq \sqrt{2}\|\Delta\|\|(\mathbb{I}_N - A - B)^{-1}\mathbf{c}\|/\widetilde{\epsilon}_B \leq \sqrt{2}\|\Delta\|\|\mathbf{c}\|/\epsilon_{AB}\widetilde{\epsilon}_B. \quad (7.15)$$

The last inequality is a consequence of the constraints **(SC)** and **(PC)**. Indeed,

$$\|(\mathbb{I}_N - A - B)^{-1}\mathbf{c}\| = \left\|\sum_{i=0}^{\infty}(A+B)^i\mathbf{c}\right\| \leq \sum_{i=0}^{\infty}\|(A+B)\|^i_{\mathrm{op}}\|\mathbf{c}\| \leq \sum_{i=0}^{\infty}(1-\epsilon_{AB})^i\|\mathbf{c}\| = \frac{\|\mathbf{c}\|}{\epsilon_{AB}}.$$

Now, by (7.10) and (7.15),

$$\|E\left[(T^*_A H_t - T^*_A H^k_t)\cdot\Delta\right]\| = \|E\left[T^*_A H_{t-k}\cdot\mathrm{vech}^*(B^{k\,T}\mathrm{math}^*(\Delta))\right]\|$$
$$\leq \|T^*_A H_{t-k}\|_{\mathrm{op}}\|\mathrm{vech}^*\|_{\mathrm{op}}\|B\|^k_{\mathrm{op}}\|\mathrm{math}^*\|_{\mathrm{op}}\|\Delta\| \leq \frac{2\|\Delta\|\|\mathbf{c}\|}{\epsilon_{AB}\widetilde{\epsilon}_B}(1-\widetilde{\epsilon}_B)^k, \quad (7.16)$$

which proves (3.18). If we want this upper bound for the error to be smaller than a certain $\delta>0$, we have to make the number of iterations k big enough so that

$$\frac{2\|\mathbf{c}\|}{\epsilon_{AB}\widetilde{\epsilon}_B}(1-\widetilde{\epsilon}_B)^k < \delta \quad \text{that is} \quad (1-\widetilde{\epsilon}_B)^k = \frac{\delta\epsilon_{AB}\widetilde{\epsilon}_B}{2\|\mathbf{c}\|} \leq \frac{\delta\epsilon_{AB}\widetilde{\epsilon}_B}{2\epsilon_{\mathbf{c}}}.$$

This relation, together with (7.14) proves the estimate (3.20). ■

References

[AB97] T. G. Andersen and T. Bollerslev. Answering the skeptics: yes, standard volatility models do provide accurate forecasts. *Journal of Empirical Finance*, 4:115–158, 1997.

[AC97] Carol Alexander and A. M. Chibumba. Multivariate orthogonal factor GARCH. *Preprint, University of Sussex*, 1997.

[Ale98] Carol Alexander. Orthogonal GARCH. In Carol Alexander, editor, *Mastering Risk*, volume 2, pages 21–38. Financial Times-Prentice Hall, 1998.

[Ale03] Carol Alexander. Principal component models for generating large covariance matrices. *Economic Notes*, 31(2):337–359, 2003.

[ASPL03] Aslıhan Altay-Salih, Mustafa Ç. Pınar, and Sven Leyffer. Constrained nonlinear programming for volatility estimation with GARCH models. *SIAM Rev.*, 45(3):485–503 (electronic), 2003.

[BEW88] Tim Bollerslev, Robert F. Engle, and J. M. Wooldridge. A capital asset pricing model with time varying covariances. *Journal of Political Economy*, 96:116–131, 1988.

[BLR06] Luc Bauwens, Sébastien Laurent, and Jeroen V. K. Rombouts. Multivariate GARCH models: a survey. *J. Appl. Econometrics*, 21(1):79–109, 2006.

[Bol86] Tim Bollerslev. Generalized autoregressive conditional heteroskedasticity. *J. Econometrics*, 31(3):307–327, 1986.

[Bol90] Tim Bollerslev. Modelling the coherence in short-run nominal exchange rates: A multivariate generalized arch model. *Review of Economics and Statistics*, 72(3):498–505, 1990.

[CGT00] Andrew R. Conn, Nicholas I. M. Gould, and Philippe L. Toint. *Trust-region methods*. MPS/SIAM Series on Optimization. Society for Industrial and Applied Mathematics (SIAM), Philadelphia, PA, 2000.

[Com94] P. Comon. Independent component analysis: a new concept? *Signal Processing*, 36:287–314, 1994.

[CR09] Emmanuel J. Candès and Benjamin Recht. Exact matrix completion via convex optimization. *Found. Comput. Math.*, 9(6):717–772, 2009.

[CT10] Emmanuel J. Candès and Terence Tao. The power of convex relaxation: near-optimal matrix completion. *IEEE Trans. Inform. Theory*, 56(5):2053–2080, 2010.

[Din94] Z. Ding. *Time Series Analysis of Speculative Returns*. PhD thesis, University of California, San Diego, 1994.

[DT07] Inderjit S. Dhillon and Joel A. Tropp. Matrix nearness problems with Bregman divergences. *SIAM J. Matrix Anal. Appl.*, 29(4):1120–1146, 2007.

[Dua95] Jin-Chuan Duan. The GARCH option pricing model. *Math. Finance*, 5(1):13–32, 1995.

[EK95] Robert F. Engle and F. K. Kroner. Multivariate simultaneous generalized arch. *Econometric Theory*, 11:122–150, 1995.

[Eng82] Robert F. Engle. Autoregressive conditional heteroscedasticity with estimates of the variance of United Kingdom inflation. *Econometrica*, 50(4):987–1007, 1982.

[Eng02] Robert F. Engle. Dynamic conditional correlation– a simple class of multivariate garch models. *Journal of Business and Economic Statistics*, 20:339–350, 2002.

[ES01] Robert F. Engle and K. Sheppard. Theoretical and empirical properties of dynamic conditional correlation multivariate garch. *Preprint, UCSD*, 2001.

[GFGPP08] A. García-Ferrer, E. González-Prieto, and Peña. A multivariate generalized independent factor garch model with an application to financial stock returns. Statistics and Econometrics Series 28, Universidad Carlos III de Madrid, 2008.

[GN86] C. Granger and P. Newbold. *Forecasting Economic Time Series*. Academic Press, San Diego, CA, second edition, 1986.

[Gou97] Christian Gouriéroux. *ARCH models and financial applications*. Springer Series in Statistics. Springer-Verlag, New York, 1997.

[HJ94] Roger A. Horn and Charles R. Johnson. *Topics in matrix analysis*. Cambridge University Press, Cambridge, 1994. Corrected reprint of the 1991 original.

[HO97] A. Hyvärinen and E. Oja. A fast fixed-point algorithm for independent component analysis. *Neural Computation*, 9:1483–1492, 1997.

[JS61] W. James and Charles Stein. Estimation with quadratic loss. In *Proc. 4th Berkeley Sympos. Math. Statist. and Prob., Vol. I*, pages 361–379. Univ. California Press, Berkeley, Calif., 1961.

[KN98] FK Kroner and VK Ng. Modelling asymmetric comovements of asset returns. *The Review of Financial Studies*, 11:817–844, 1998.

[KSD09a] Brian Kulis, Mátyás A. Sustik, and Inderjit S. Dhillon. Low-rank kernel learning with Bregman matrix divergences. *J. Mach. Learn. Res.*, 10:341–376, 2009.

[KSD09b] Brian Kulis, Suvrit S. Sustik, and Inderjit S. Dhillon. Convex perturbations for scalable semidefinite programming. In *Proceedings of the 12th International Conference on Artificial Intelligence and Statistics (AISTATS) 2009*, pages 296–303, 2009.

[Lüt05] Helmut Lütkepohl. *New introduction to multiple time series analysis*. Springer-Verlag, Berlin, 2005.

[MCV02] Simone Manganelli, Vladimiro Ceci, and Walter Vecchiato. Sensitivity Analysis of Volatility: A New Tool for Risk Management. *European Central Bank, Working Paper No. 194*, 2002.

[MN79] Jan R. Magnus and H. Neudecker. The commutation matrix: some properties and applications. *Ann. Statist.*, 7(2):381–394, 1979.

[MZ69] J. Mincer and V. Zarnowitz. The evaluation of economic forecasts. In J. Mincer, editor, *Economics Forecasts and Expectations*, New York, 1969. National Bureau of Economic Research.

[Ris96] Riskmetrics. *Riskmetrics Technical Document*. J. P. Morgan, New York, 4th edition, 1996.

[ST09] A. Silvennoinen and T. Teräsvirta. Multivariate garch models. In *Handbook of Financial Time Series*, pages 201–229. Springer, Berlin, 2009.

[TT02] Y. K. Tse and A. K. C. Tsui. A multivariate garch with time-varying correlations. *Journal of Business and Economic Statistics*, 20:351–362, 2002.

[vdW02] Roy van der Weide. Go-garch: a multivariate generalized orthogonal garch model. *J. Appl. Econ.*, 17(17):549–564, 2002.

[WYL06] Edmond H. C. Wu, Philip L. H. Yu, and W. K. Li. Value at risk estimation using independent component analysis-generalized autoregressive conditional heteroscedasticity (ica-garch) models. *International Journal of Neural Systems*, 16(5):371–382, 2006.